

**APLIKASI PENGENALAN SUARA SEBAGAI
PENGENDALI PERALATAN LISTRIK BERBASIS
ARDUINO UNO**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

ZARATUL NISA SAPUTRI

NIM.105060300111052-63

KEMENTERIAN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2014

LEMBAR PERSETUJUAN

**APLIKASI PENGENALAN SUARA SEBAGAI PENGENDALI
PERALATAN LISTRIK BERBASIS ARDUINO UNO**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

ZARATUL NISA SAPUTRI

NIM.105060300111052-63

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Moch. Rif'an, ST., MT.

NIP. 19710301 200012 1 001

Ir. Nurussa'adah, MT.

NIP. 19680706 199203 2 001

LEMBAR PENGESAHAN

**APLIKASI PENGENALAN SUARA SEBAGAI PENGENDALI
PERALATAN LISTRIK BERBASIS ARDUINO UNO**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

ZARATUL NISA SAPUTRI

NIM.105060300111052-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 11 Agustus 2014

Telah diperiksa dan disetujui oleh:

Penguji

Penguji

Ir. M.Julius,ST., MS.

NIP. 19540720 198203 1 002

Ir. Ponco Siwindarto, M.Eng.Sc

NIP. 19590304 198903 1 001

Penguji

Ir. Nanang Sulistyanto, M.T

NIP. 19700113 199403 1 002

Mengetahui

Ketua Jurusan Teknik Elektro

M. Aziz Muslim, ST., MT., Ph.D

NIP. 19741203 200012 1 001

PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Aplikasi Pengenalan Suara Sebagai Pengendali Peralatan Listrik Berbasis *ArduinoUNO*” dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar Sarjana Teknik dari jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Penulis menyadari bahwa tanpa bantuan, bimbingan serta dorongan dari semua pihak, penyelesaian skripsi ini tidak mungkin bisa terwujud. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

- Bapak Suyatno Hadi Saputro dan Ibu Siti Maisarah selaku orang tua atas limpahan kasih sayang, dukungan moril dan materiil, serta curahan doa yang tidak pernah putus hingga skripsi ini dapat diselesaikan.
- Bapak Azis Muslim, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya
- Bapak Hadi Suyono, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro universitas Brawijaya
- Bapak Mochammad Rif'an, ST., MT. selaku Ketua Prodi Strata Satu Jurusan Teknik Elektro Universitas Brawijaya dan juga Dosen pembimbing I penulis atas segala pengarahan ide, saran, dan bimbingan yang telah diberikan
- Ibu Dr. Ing. Onny Setyawati, ST., MT. selaku Dosen Pembimbing akademik atas segala bimbingan, motivasi dan masukan yang telah diberikan.
- Ibu Ir. Nurussa'adah, MT. sebagai Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya dan juga Dosen Pembimbing II atas segala bimbingan, pengarahan ide, nasihat, saran dan kritik yang telah diberikan
- Bapak dan Ibu Dosen Jurusan Teknik Elektro

- Staff Recording Jurusan Teknik Elektro
- Lucky Meritya Fatma sebagai sahabat terbaik atas dukungan semangat, perhatian, dan doa sehingga penulis lebih termotivasi menyelesaikan skripsi ini.
- Mas Jatra Kurnia Ardi dan Nur Iskandar juang yang telah banyak membantu dalam penyelesaian skripsi ini.
- Rekan seperjuangan Tanshuda Al fauzi dan Naufal Anas terima kasih atas segala bantuan yang telah diberikan
- Teman-teman Magnety Reni, Nilfa, Rita, Safti, Rara, Ayu, Resi, Nana, Tika, Reta, Ika, Anne, Imee, Dina, Neta, Mega, Shanty, Nunik, Erny, Dewi, Aya, Anastasia, dan Mitha.
- Teman-teman sepermainan, Azis, Davin, Ario, Topher dan Enov terima kasih telah mengajarkan cara menikmati dan bersyukur lebih banyak akan hidup.
- Teman-teman Golden House kav. 04 Rahma, Sukma, Dara, Kak Tiara, Kak Dini, Riana, Taufanni dan Afi.
- Teman-teman Magnet Angkatan 2010
- Seluruh teman-teman serta semua pihak yang tidak mungkin untuk dicantumkan satu persatu, terima kasih banyak atas segala bentuk bantuan dan dukungannya.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaan skripsi ini. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Agustus 2014

Penulis

DAFTAR ISI

PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
ABSTRAK	ix
PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Runusan Masalah.....	1
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	2
1.5. Sistematika Penulisan.....	2
TINJAUAN PUSTAKA	4
2.1. Pendekatan Pengenalan Pola.....	4
2.2. Sistem Analisis Ucapan.....	5
2.3. Metode MFCC (<i>Mel Frequency Cepstrum Coefficients</i>).....	5
2.3.1. Konversi Sinyal Analog menjadi Digital.....	7
2.3.2. Remove DC.....	8
2.3.3. <i>Pre-Emphasis Filter</i>	9
2.3.4. <i>Frame Blocking</i>	10
2.3.5. Windowing.....	11
2.3.6. Analisis Fourier.....	12
2.3.7. <i>Mel Frequency Wrapping</i>	12
2.3.8. <i>Discrete Cosine Transform (DCT)</i>	14
2.3.9. <i>Cepstral Liftering</i>	14

2.4. Modul <i>EasyVR</i>	15
2.4.1. Fungsi Pin pada <i>EasyVR</i>	16
2.4.2. Karakteristik Elektrik <i>EasyVR</i>	17
2.4.3. <i>Communication Protocol</i>	17
2.4.4. <i>EasyVR Commander</i>	18
2.5. Sensor <i>Michrophone</i>	19
2.5.1. Jenis-jenis <i>Michrophone</i> berdasarkan Kepekaannya.....	20
2.5.2. Sifat-sifat <i>Michrophone</i>	20
2.6. <i>Arduino UNO</i>	20
2.6.1. Catu Daya.....	21
2.6.2. <i>Memory</i>	22
2.6.3. <i>Input&Output</i>	22
2.6.4. Komunikasi.....	23
2.6.5. <i>Programming</i>	24
2.6.6. Perangkat Lunak (<i>Arduino IDE</i>).....	24
2.6.7. Otomatis <i>Software Reset</i>	24
2.7. <i>Relay</i>	25
2.8. Transistor sebagai Saklar.....	25
2.9. Komunikasi Serial.....	28
2.10. Komunikasi <i>Wireless Microphone</i>	29
METODOLOGI PENELITIAN	31
3.1. Penentuan Spesifikasi Alat.....	31
3.2. Studi Literatur.....	31
3.3. Perancangan <i>Voice Recognition</i> Menggunakan <i>EasyVR</i>	32
3.3.1. Perancangan Sistem Komunikasi <i>EasyVR</i> ke Komputer.....	32
3.3.2. Perancangan Sistem Komunikasi <i>EasyVR</i> ke <i>Arduino UNO</i>	32



3.3.3. Perancangan Pengambilan <i>Sample</i> Suara Menggunakan <i>EasyVR</i>	
<i>Commander</i>	33
3.4. Perancangan dan Penyusunan Perangkat Lunak.....	34
3.4.1. Perancangan Sistem Secara Keseluruhan.....	35
3.5. Pengujian Alat.....	36
3.5.1. Pengujian Tiap Blok.....	36
3.5.2. Pengujian Keseluruhan Sistem.....	36
3.6. Pengambilan Kesimpulan.....	36
PERANCANGAN DAN PEMBUATAN ALAT.....	38
4.1. Perancangan Sistem.....	38
4.2. Perancangan Perangkat Keras.....	39
4.2.1. Perancangan Desain Sistem Elektronik.....	39
4.2.2. Perancangan Catu Daya Sistem.....	40
4.2.3. Perancangan Rangkaian Relay.....	43
4.2.4. Perancangan Rangkaian Mikrokontroler Pengatur Utama.....	45
4.3. Hubungan <i>Arduino UNO</i> dengan <i>EasyVR</i>	45
4.3.1. Perancangan Sistem Pengenalan Suara Menggunakan <i>EasyVR</i>	46
4.4. Perancangan dan Pembuatan Perangkat Lunak (<i>Software</i>).....	48
4.4.1. Perancangan Susunan Perangkat Lunak.....	48
4.4.2. Diagram Alir Keseluruhan Sistem.....	49
4.4.3. Diagram Alir Program Pendeteksi <i>EasyVR</i>	50
PENGUJIAN DAN ANALISIS.....	51
5.1. Pengujian Komunikasi Serial UART ke PC.....	51
5.2. Pengujian <i>Voice Recognition</i>	52
5.2.1. Pengujian Hasil <i>Sampling</i>	52
5.2.2. Pengujian Tingkat Keberhasilan Pemberian Perintah dengan	

Variasi Pemberian Perintah dari Orang yang Sama.....	54
5.2.3. Pengujian Pemberian Perintah dari Orang yang Berbeda.	62
5.3. Pengujian Keberhasilan Menerima Perintah Dengan Jarak Jauh Menggunakan <i>Wireless Microphone</i>	63
5.4. Pengujian Keseluruhan Sistem.....	64
5.4.1. Pengujian Melalui Serial Monitor pada Program <i>Arduino</i>	64
5.4.2. Pengujian Respon Peralatan Rumah Tangga Setelah Menerima Perintah Dalam Satuan Waktu.....	66
PENUTUP	67
6.1. Kesimpulan.....	67
6.2. Saran.....	67
DAFTAR PUSTAKA	68
LAMPIRAN 1	69
LAMPIRAN 2	72
LAMPIRAN 3	80



DAFTAR GAMBAR

Gambar 2.1. Blok diagram pendekatan dengan pengenalan pola.	4
Gambar 2.2. Contoh Sinyal Suara	6
Gambar 2.3 Blok diagram MFCC.	7
Gambar 2.4. Proses Pembentukan Sinyal Digital.	8
Gambar 2.5. Contoh dari <i>Pre-Emphasis</i> pada sebuah <i>Frame</i>	9
Gambar 2.6. Bentuk Sinyal yang di <i>Frame Blocking</i>	11
Gambar 2.7. <i>Magnitude</i> dari <i>Rectangulare</i> dan <i>Tringular Filterbank</i>	13
Gambar 2.8. Konfigurasi pin pada <i>EasyVR</i>	16
Gambar 2.9. Protokol komunikasi pada modul <i>EasyVR</i>	17
Gambar 2.10. Tampilan pada <i>EasyVR Commander</i>	19
Gambar 2.11. Board <i>Arduino UNO</i> dan kabel USB.	21
Gambar 2.12 Tampilan <i>framework Arduino</i>	24
Gambar 2.13 Grafik Kurva Karakteristik Transistor.	26
Gambar 2.14 Transistor Kondisi Jenuh (saklar posisi <i>ON</i>).	27
Gambar 2.15. Transistor Kondisi Mati (saklar posisi <i>OFF</i>).	28
Gambar 2.16. Format frame data serial <i>USART</i>	29
Gambar 2.17. Blok diagram rangkaian <i>wireless microphone</i>	30
Gambar 3.1. Proses Komunikasi <i>EasyVR</i> dengan komputer..	32
Gambar 3.2. Proses komunikasi <i>EasyVR</i> dengan <i>Arduino UNO</i>	32
Gambar 3.3. Blok diagram sistem <i>voice reognition</i> melalui <i>EasyVR</i>	33
Gambar 3.4. Tampilan pada <i>software EasyVR Commander</i>	33
Gambar 3.5. Tampilan pada <i>software EasyVR Commander</i>	34
Gambar 3.6. Diagram blok sistem secara keseluruhan.	35
Gambar 4.1. Diagram Blok Sistem.	38
Gambar 4.2. Rangkain Catu Daya.	40
Gambar 4.3. Rangkaian <i>Relay</i> Mekanik.	43
Gambar 4.4. Rangkaian RLC.	43
Gambar 4.5. Transistor pada <i>relay</i>	44
Gambar 4.6. Minimum Sistem <i>Arduino UNO</i>	45
Gambar 4.7. Minimum Sistem <i>EasyVR</i>	46
Gambar 4.8. Diagram blok pemberian <i>sample</i> suara.	46
Gambar 4.9. Tampilan kegagalan dalam pengambilan <i>sample</i> suara.	47
Gambar 4.10 Sampling Suara Pada <i>EasyVR Commander</i>	47
Gambar 4.11 Proses Penyamplingan suara menggunakan <i>EasyVR</i>	48
Gambar 4.12. Diagram alir keseluruhan sistem.	49
Gambar 4.13. Diagram alir program pendeteksi <i>EasyVR</i>	50
Gambar 5.1. Tampilan Serial Monitor pada <i>Arduino</i> untuk pengujian komunikasi serial dengan USB.	51
Gambar 5.2. Hasil Pengujian <i>Sampling</i>	54
Gambar 5.3. Tampilan serial Monitor untuk pengujian peralatan listrik (a) Lampu Teras menyala dan padam, (b) Kipas Menyala, (c) Kipas padam, (d) TV menyala, (e) Tv padam.	65

DAFTAR TABEL

Tabel 2.1. Fungsi pin <i>EasyVR</i>	16
Tabel 2.2. Karakteristik elektrik <i>EasyVR</i>	17
Tabel 2.3. Karakteristik <i>Arduino Uno</i>	21
Tabel 5.1. Hasil pengujian pemberian perintah sesuai dengan perancangan.....	54
Tabel 5.2. Hasil pengujian pemberian perintah dengan menggunakan 2 kata dalam setiap kalimat.....	55
Tabel 5.3. Hasil pengujian pemberian perintah menggunakan kode <i>Alphabet</i>	55
Tabel 5.4 Besar Tegangan Lampu pada Perintah “Nyalakan Lampu Teras”	56
Tabel 5.5 Besar Tegangan Lampu pada Perintah “Teras Padam”	56
Tabel 5.6 Besar Tegangan Kipas Angin pada Perintah “Hidupkan Fan”	57
Tabel 5.7 Besar Tegangan Kipas Angin pada Perintah “Matikan Kipas Angin”	57
Tabel 5.8 Besar Tegangan Televisi pada Perintah “TV ON”	57
Tabel 5.9 Besar Tegangan Televisi pada Perintah “OFFkan TV”	57
Tabel 5.10 Besar Tegangan Lampu pada Perintah “Lampu Nyalala”	58
Tabel 5.11 Besar Tegangan Lampu pada Perintah “Padam Lampu”	58
Tabel 5.12 Besar Tegangan Kipas angin pada Perintah “Kipas ON”	58
Tabel 5.13 Besar Tegangan Kipas angin pada Perintah “Mati Kipas”	59
Tabel 5.14 Besar Tegangan Televisi pada Perintah “TV Hidup”	59
Tabel 5.15 Besar Tegangan Televisi pada Perintah “OFF TV”	59
Tabel 5.16 Besar Tegangan Lampu pada Perintah “A”	60
Tabel 5.17 Besar Tegangan Lampu pada Perintah “B”	60
Tabel 5.18 Besar Tegangan Kipas Angin pada Perintah “C”	60
Tabel 5.19 Besar Tegangan Kipas Angin pada Perintah “D”	61
Tabel 5.20 Besar Tegangan Televisi pada Perintah “E”	61
Tabel 5.21 Besar Tegangan Televisi pada Perintah “F”	61
Tabel 5.22. Hasil pengujian pemberian perintah dari orang yang berbeda.....	62
Tabel 5.23. Hasil Pengujian keberhasilan menerima perintah dengan jarak jauh menggunakan <i>wireless microphone</i>	63
Tabel 5.24 Hasil Pengujian rata-rata waktu respon peralatan listrik setelah menerima Perintah.....	66

ABSTRAK

Zaratul Nisa Saputri, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juli 2014, Aplikasi Pengenalan Suara Sebagai Pengendali Peralatan Listrik Berbasis *ArduinoUNO*. Dosen Pembimbing: Mochammad Rif'an, ST., MT. dan Ir. Nurussa'adah, MT.

Perkembangan ilmu pengetahuan dan teknologi membawa dampak positif dalam kehidupan manusia yang pada saat ini telah sampai pada zaman perintah suara. Sistem kontrol rumah pintar memungkinkan manusia mengendalikan perangkat listrik rumah mereka seperti TV, kipas angin dan lampu hanya dengan menggunakan perintah suara tanpa perlu bergerak berpindah tempat untuk menyalakan atau mematikan suatu peralatan listrik. Pengolahan suara digital dikontrol dengan aplikasi untuk mengenali adanya perintah suara yang dideteksi.

Oleh karena itu dalam skripsi ini akan dirancang Aplikasi Pengenalan Suara Sebagai Pengendali Peralatan Listrik Berbasis *ArduinoUNO*. Digunakan modul *EasyVR* sebagai modul pengenalan suara. Proses pengambilan *sample* suara pada *EasyVR* dilakukan sebanyak dua kali dengan variasi pengucapan relatif sama pada setiap kata. Hal ini dilakukan sesuai dengan kemampuan *EasyVR* yang tidak bisa menerima pengucapan variasi suara kedua jika berbeda dengan variasi pengucapan suara pertama.

Penerapan aplikasi pengenalan suara pada peralatan listrik ini menggunakan *Relay* yang berfungsi sebagai saklar, digunakan *microphone wireless* agar pengucapan dapat dilakukan dari jarak jauh.

Kata Kunci: Pengenalan Suara, *EasyVR*, *ArduinoUNO*.

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Perkembangan ilmu pengetahuan dan teknologi membawa dampak positif dalam kehidupan manusia yang pada saat ini telah sampai pada zaman perintah suara. Sistem kontrol rumah pintar memungkinkan orang mengendalikan perangkat rumah mereka dengan perintah suara di rumah. Pengolahan suara digital dapat dikembangkan untuk mempermudah kehidupan manusia. Dalam hal ini suara manusia dapat diolah untuk dikonversi agar dimengerti oleh suatu responden sehingga perintah yang terucap dapat direspon oleh alat yang dikendalikan. Salah satu yang dapat dibuat adalah aplikasi untuk pengendalian jarak jauh.

Pengolahan suara digital dikontrol dengan aplikasi untuk mengenali adanya perintah suara yang dideteksi, yang sering disebut dengan *Speech Recognition*. Teknologi ini bekerja dengan menangkap suara manusia yang diubah menjadi format digital sehingga dapat diterjemahkan dalam suatu sistem. Kemudian sistem tersebut akan membandingkan antara informasi masukkan yang sudah berupa format digital tersebut dengan *database* suara yang ada (Ardi, 2014).

Sistem suara nirkabel, mengontrol beberapa fasilitas di rumah seperti lampu, kipas angin dan televisi. Fungsi kontrol suara ini adalah sebagai pengendali peralatan listrik rumah untuk menyalakan, mematikan dan menggantikan fungsi tombol dengan suara. Perkembangan kontrol suara pengendali rumah dengan teknologi *wireless* ini bisa menjadi sebagai acuan rumah masa depan, di mana kita tidak perlu lagi berpindah tempat hanya untuk menyalakan dan mematikan suatu peralatan listrik yang ada di suatu tempat yang berbeda didalam rumah kita.

1.2 RUMUSAN MASALAH

Berdasarkan latar belakang tersebut, dapat disusun rumusan masalah sebagai berikut:

- 1) Bagaimana merancang sistem *voice recognition* menggunakan modul *EasyVR*.
- 2) Bagaimana tingkat keberhasilan metode sistem *voice recognition* jika diukur dari pemberian perintah dari orang yang berbeda.
- 3) Bagaimana efektifitas modul *EasyVR* dalam proses pemberian perintah.

1.3 BATASAN MASALAH

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan penelitian akan diberi batasan sebagai berikut:

- 1) Mikrokontroler yang dipakai adalah *Arduino UNO* sebagai pengendali utama.
- 2) Modul *voice recognition* yang digunakan adalah *EasyVR*.
- 3) Perintah yang digunakan adalah *ON* dan *OFF* pada 3 jenis peralatan listrik rumah tangga seperti lampu, televisi dan kipas angin.

1.4 TUJUAN

Tujuan penelitian ini adalah merancang dan membuat sistem *voice recognition* sebagai sistem pengendali peralatan listrik rumah tangga yang dapat dikendalikan dari jarak jauh menggunakan *michrophone wireless* yang ada di pasaran.

1.5 SISTEMATIKA PENULISAN

Sistematika penulisan dalam penelitian ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, ruang lingkup, tujuan, dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Metodologi

Berisi tentang spesifikasi alat, metode-metode yang dipakai dalam pengujian, sehingga variasi data yang dihasilkan dari pengujian dapat ditarik kesimpulan.

BAB IV Perancangan

Perancangan dan perealisasiian alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan realisasi alat.

BAB V Pengujian dan Analisis

Memuat aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar terhadap hasil-hasil pengujian. Pengujian dan analisis ini terhadap alat yang telah direalisasikan berdasarkan masing-masing blok dan sistem secara keseluruhan.

BAB VI Kesimpulan dan Saran

Memuat intisari hasil pengujian dan menjawab rumusan masalah serta memberikan rekomendasi untuk perbaikan kualitas penelitian dimasa yang akan datang.



BAB II

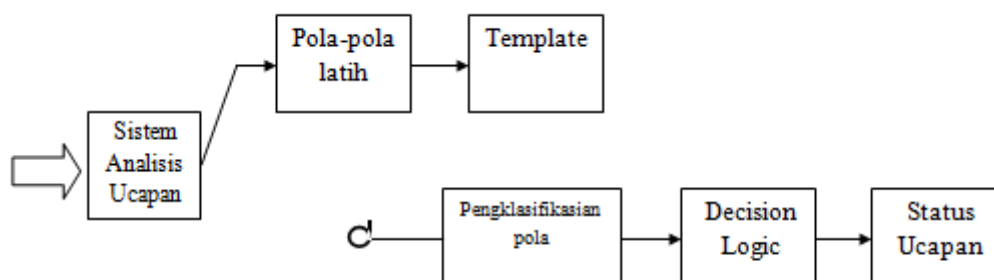
TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan sistem yang dibuat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penelitian ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah:

- Pengenalan Ucapan Manusia
- Modul *EasyVR*
- *Arduino UNO*
- *Relay*
- Transistor sebagai Saklar
- Komunikasi serial
- *Microphone Wireless*

2.1 Pendekatan Pengenalan Pola

Gambar 2.1 menunjukkan blok diagram sistem pengenalan ucapan dengan pendekatan pengenalan pola. Blok sistem analisis ucapan berfungsi untuk mendapatkan *feature* sinyal suara yang berupa suatu representasi spektral. Pada sistem ini dapat digunakan beberapa metode analisis ucapan seperti *Bank-of-filter*, *Linear Predictive Coding* atau transformasi frekuensi. Kemudian keluaran sistem yang berupa pola-pola digunakan sebagai pola latihan untuk dikenali (Rabiner dan Juang 1993).



Gambar 2.1 Blok diagram pendekatan dengan pengenalan pola

Sumber: Rabiner dan Juang, (1993)

Pada proses pelatihan, pola-pola tersebut diproses dengan suatu algoritma untuk mendapatkan pola-pola referensi yang disebut sebagai *template*. Sedangkan pada proses pengenalan pola-pola ucapan yang merupakan keluaran blok sistem analisis ucapan diklasifikasikan dengan cara mengukur distorsi pola-pola ucapan tersebut terhadap sumber pola-pola referensi. Kemudian distorsi tadi akan dianalisis oleh *Decision Logic* untuk diputuskan apakah ucapan tersebut dikenali atau tidak.

2.2 Sistem Analisis Ucapan

Sistem analisis ucapan merupakan sistem yang sangat penting pada setiap metoda pendekatan pengenalan ucapan. Sistem ini berfungsi untuk mengubah gelombang suara (*speech wave*) menjadi representasi parameter (pernyataan matematika dari suatu ucapan) tertentu untuk digunakan pada pengolahan berikutnya.

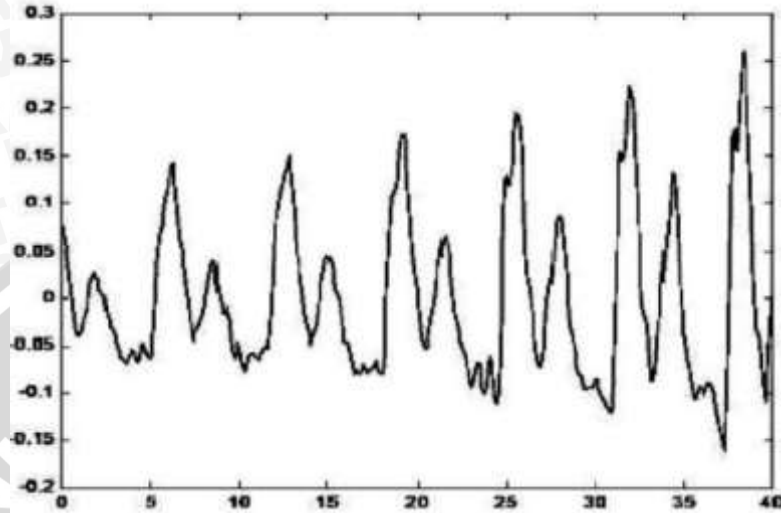
Metoda yang biasa digunakan dalam analisis ucapan ini akan mengekstraksi suara manusia sehingga didapatkan sederetan *feature* ucapan yang umumnya berupa representasi spektral dan karakteristik sinyal masukan yang berubah terhadap waktu. Kumpulan *feature* tersebut kemudian menjadi masukan pengklasifikasi pola.

2.3 Metode MFCC (*Mel Frequency Cepstrum Coefficients*)

MFCC (*Mel Frequency Cepstrum Coefficients*) merupakan salah satu metode yang banyak digunakan dalam bidang *speech technology*, baik *speaker recognition* maupun *speech recognition*. Metode ini digunakan untuk melakukan *feature extraction*, sebuah proses yang mengkonversikan sinyal suara menjadi beberapa parameter. Beberapa keunggulan dari metode ini adalah sebagai berikut:

- a. Mampu untuk menangkap karakteristik suara yang sangat penting bagi pengenalan suara, atau dengan kata lain dapat menangkap informasi-informasi penting yang terkandung dalam sinyal suara.
- b. Menghasilkan data seminimal mungkin, tanpa menghilangkan informasi-informasi penting yang dikandungnya.

- c. Mereplikasi organ pendengaran manusia dalam melakukan persepsi terhadap sinyal suara.

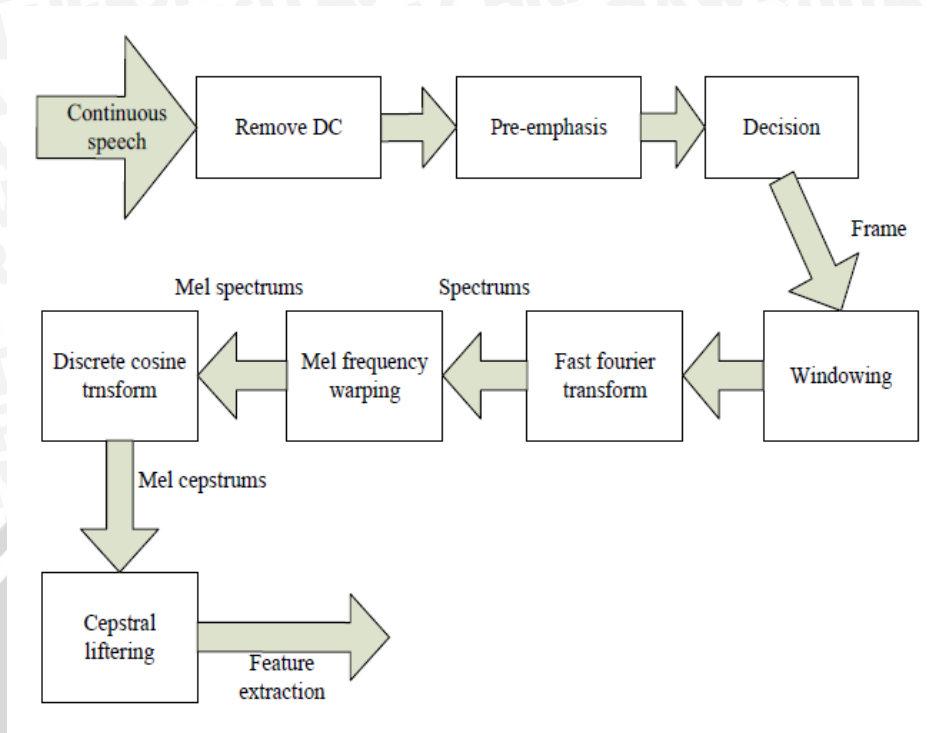


Gambar 2.2. Contoh Sinyal Suara

Sumber: Putra, (2009)

Contoh dari sinyal suara dapat dilihat pada Gambar 2.2 pengujian yang dilakukan untuk periode waktu yang cukup pendek (sekitar 10 sampai 30 milidetik) akan menunjukkan karakteristik sinyal suara yang *stationary*. Tetapi bila dilakukan dalam periode waktu yang lebih panjang karakteristik *signal* suara akan terus berubah sesuai dengan kata yang diucapkan.

MFCC *feature extraction* sebenarnya merupakan adaptasi dari sistem pendengaran manusia, dimana *signal* suara akan di-*filter* secara linear untuk frekuensi rendah (dibawah 1000 Hz) dan secara logaritmik untuk frekuensi tinggi (diatas 1000 Hz). Gambar 2.3 merupakan blok diagram untuk MFCC.



Gambar 2.3 Blok diagram MFCC

Sumber: Putra, (2009)

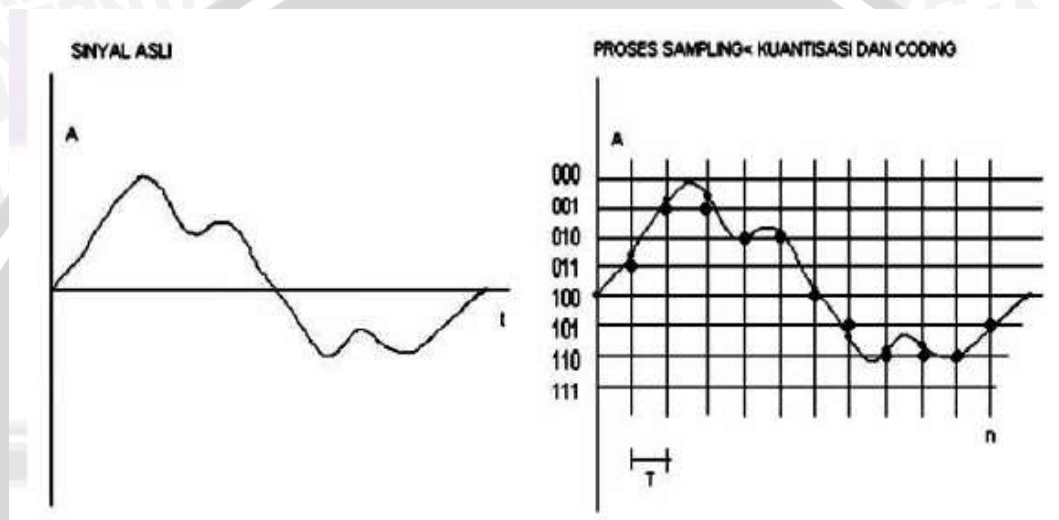
2.3.1. Konversi Sinyal *Analog* menjadi *Digital*

Sinyal–sinyal yang natural pada umumnya seperti sinyal suara merupakan *signal continue* dimana memiliki nilai yang tidak terbatas. Sedangkan pada komputer, semua sinyal yang dapat diproses oleh komputer hanyalah *signal discrete* atau sering dikenal sebagai istilah *digital signal*. Agar sinyal natural dapat diproses oleh komputer, maka harus diubah terlebih dahulu dari data *signal continue* menjadi *discrete*. Hal itu dapat dilakukan melalui 3 proses, diantaranya adalah proses *sampling* data, proses kuantisasi dan proses pengkodean.

Proses *sampling* adalah suatu proses untuk mengambil data *signal continue* untuk setiap periode tertentu. Dalam melakukan proses *sampling* data, berlaku aturan Nyquist, yaitu bahwa frekuensi *sampling* (*sampling rate*) minimal harus 2 kali lebih tinggi dari frekuensi maksimum yang akan di-*sampling*. Jika *signal sampling* kurang dari 2 kali frekuensi maksimum sinyal yang akan di-*sampling*, maka akan timbul efek *aliasing*. *Aliasing* adalah suatu efek dimana sinyal yang dihasilkan memiliki frekuensi yang berbeda dengan sinyal aslinya.

Proses kuantisasi adalah proses untuk membulatkan nilai data ke dalam bilangan-bilangan tertentu yang telah ditentukan terlebih dahulu. Semakin banyak level yang dipakai maka semakin akurat pula data sinyal yang disimpan tetapi akan menghasilkan ukuran data besar dan proses yang lama.

Proses pengkodean adalah proses pemberian kode untuk tiap-tiap data sinyal yang telah terkuantisasi berdasarkan level yang ditempati.



Gambar 2.4. Proses Pembentukan Sinyal Digital

Sumber: Putra, (2009)

2.3.2. Remove DC

Remove DC bertujuan untuk menghitung rata-rata dari data sampel suara, dan mengurangi nilai setiap sampel suara dengan nilai rata-rata tersebut. Tujuannya adalah mendapat normalisasi dari data suara *input*. Secara matematis dapat ditulis (Putra 2009:37):

$$y[n] = x[n] - \bar{x}, 0 \leq n \leq N - 1 \quad (2-1)$$

dimana:

$y[n]$ = sampel sinyal hasil proses *remove DC*.

$x[n]$ = sampel sinyal asli.

\bar{x} = nilai rata-rata sampel sinyal asli.

N = panjang sinyal

2.3.3 Pre-Emphasis Filter

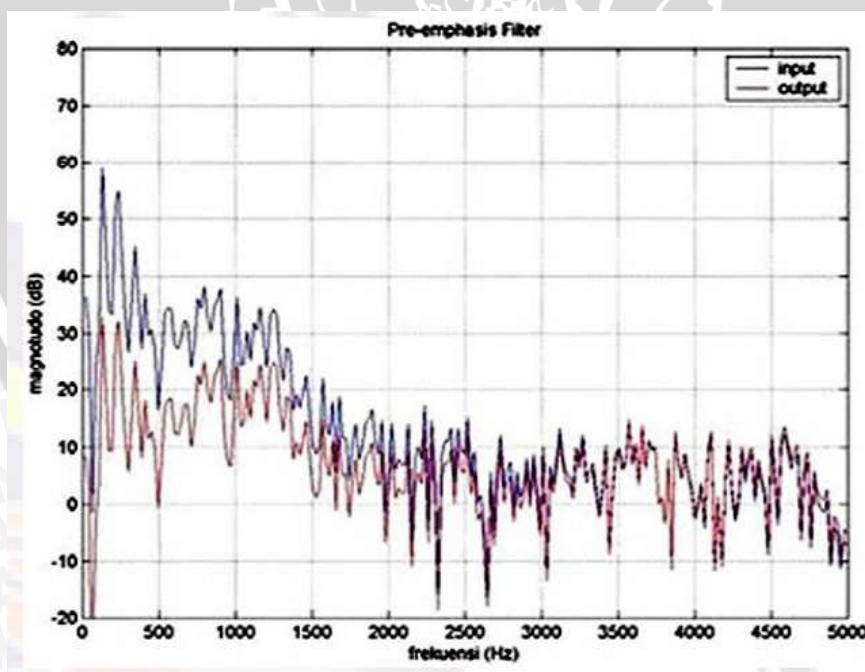
Pre-emphasis filter merupakan salah satu jenis *filter* yang sering digunakan sebelum sebuah sinyal diproses lebih lanjut. *Filter* ini mempertahankan frekuensi-frekuensi tinggi pada sebuah *spectrum*, yang umumnya tereliminasi pada saat proses produksi suara.

Tujuan dari *Pre-emphasis filter* ini adalah sebagai berikut:

a. Mengurangi *noise ratio* pada sinyal, sehingga dapat meningkatkan kualitas sinyal.

b. Menyeimbangkan spektrum dari *voice sound*. Pada saat memproduksi *voiced sound*, *glottis* manusia

menghasilkan sekitar -12 dB *octave slope*. Namun ketika energi akustik tersebut dikeluarkan melalui bibir, terjadi peningkatan sebesar $+6$ dB. Sehingga sinyal yang terekam oleh *microphone* adalah sekitar -6 dB *octave slope*. Dampak dari hal ini ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Contoh dari *Pre-Emphasis* pada sebuah *Frame*

Sumber: Putra, (2009)

Pada Gambar 2.5 terlihat bahwa distribusi energi pada setiap frekuensi terlihat lebih seimbang setelah diimplementasikan *pre-emphasis filter*. Bentuk yang paling umum digunakan dalam *pre-emphasis filter* adalah sebagai berikut: (Putra, 2009:38):

$$H(z) = 1 - \alpha z^{-1} \quad (2-2)$$

Dimana $0.9 \leq \alpha \leq 1.0$, dan $\alpha \in R$. Persamaan diatas dapat dijadikan sebagai *first order differentiator*, sebagai berikut:

$$y[n] = s[n] - \alpha s[n - 1] \quad (2-3)$$

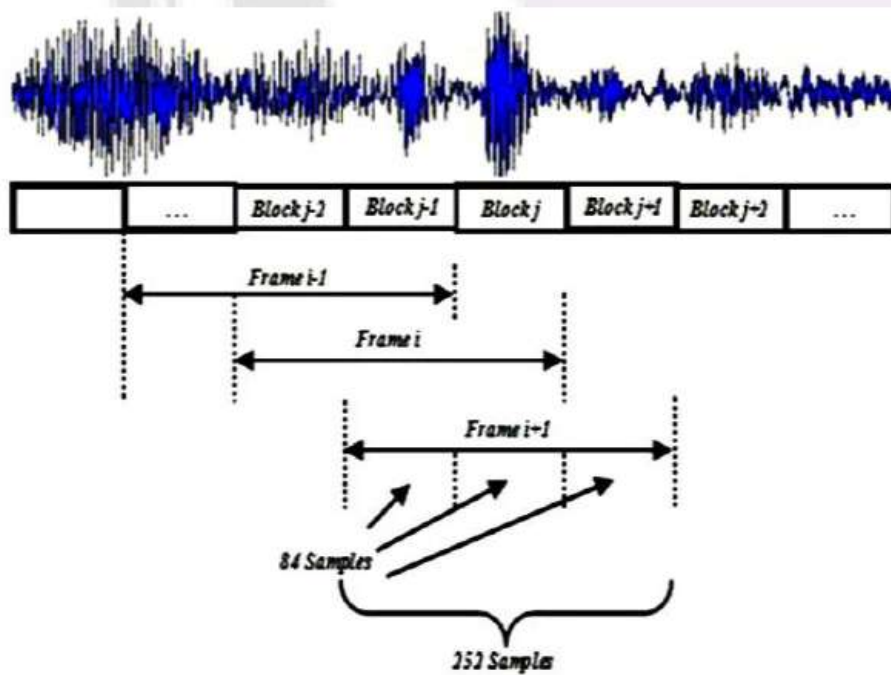
dimana:

$y[n]$ = sinyal hasil *pre-emphasis filter*.

$s[n]$ = sinyal sebelum *pre-emphasis filter*.

2.3.4. Frame Blocking

Karena sinyal suara terus mengalami perubahan akibat adanya pergeseran artikulasi dari organ produksi vokal, sinyal harus diproses secara *short segments* (*short frame*). Panjang *frame* yang biasa digunakan untuk pemrosesan sinyal adalah antara 10-30 milidetik. Panjang *frame* yang digunakan sangat mempengaruhi keberhasilan dalam analisa spektral. Di satu sisi, ukuran dari *frame* harus sepanjang mungkin untuk dapat menunjukkan frekuensi yang baik. Tetapi di lain sisi, ukuran *frame* juga harus cukup pendek untuk dapat menunjukkan resolusi waktu yang baik.



Gambar 2.6 Bentuk Sinyal yang di *Frame Blocking*

Sumber: Putra, (2009)

Proses *frame* ini dilakukan terus sampai seluruh sinyal dapat diproses. Selain itu, proses ini umumnya dilakukan secara *overlapping* untuk setiap *frame*-nya. Panjang daerah *overlap* yang umum digunakan adalah kurang lebih 30% sampai 50% dari panjang *frame*. *Overlapping* dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap *frame*.

2.3.5. *Windowing*

Proses *framing* dapat menyebabkan terjadinya kebocoran spektral (*spectral leakage*) atau *aliasing*. *Aliasing* adalah sinyal baru dimana memiliki frekuensi yang berbeda dengan sinyal aslinya. Efek ini dapat terjadi karena rendahnya jumlah *sampling rate*, ataupun karena proses *frame blocking* dimana menyebabkan sinyal menjadi *discontinue*. Untuk mengurangi kemungkinan terjadinya kebocoran spektral, maka hasil dari proses *framing* harus melewati proses *window*.

Sebuah fungsi *window* yang baik harus menyempit pada bagian *main lobe* dan melebar pada bagian *side lobe*-nya.

Berikut adalah representasi dari fungsi *window* terhadap sinyal suara yang diinputkan. (Putra, 2009:38):

$$x(n) = x_t(n)w(n), n = 0, 1, \dots, (N - 1) \quad (2-4)$$

dimana:

$x(n)$ = nilai sampel sinyal hasil *windowing*

x_t = nilai sampel dari *frame* sinyal ke i

$w(n)$ = fungsi *window*

N = *Frame size*, merupakan kelipatan 2.

Terdapat banyak fungsi *window*, namun yang paling sering digunakan dalam aplikasi *speaker recognition* adalah *hamming window*. Fungsi *window* ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang lebih -43 dB), selain itu *noise* yang dihasilkan pun tidak terlalu besar. Fungsi *Hamming window* adalah sebagai berikut: (Putra, 2009:39):

$$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{M-1} \quad (2-5)$$

dimana:

n = $0, 1, \dots, M - 1$

M = panjang *frame*

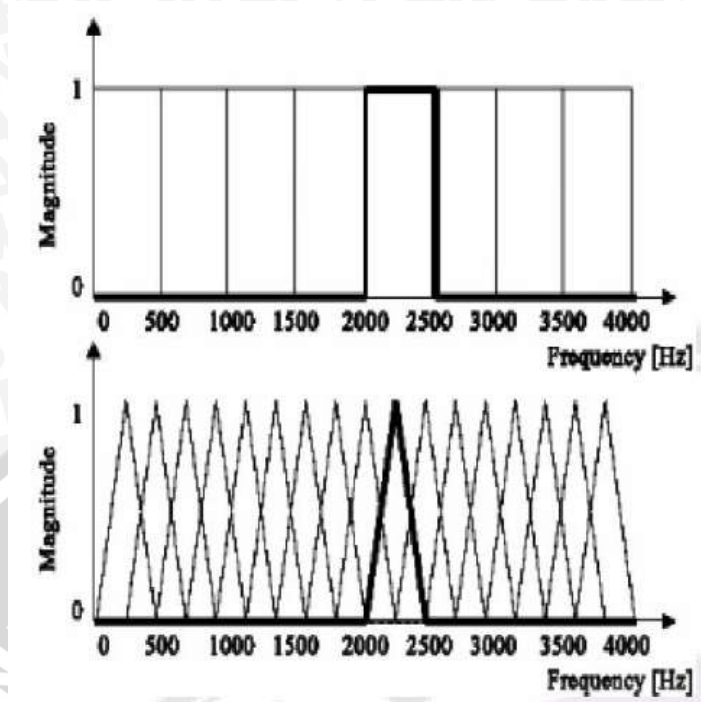
2.3.6 Analisis Fourier

Analisis *fourier* adalah sebuah metode yang memungkinkan untuk melakukan analisa terhadap *spectral properties* dari sinyal yang diinputkan. Representasi dari *spectral properties* sering disebut sebagai *spectrogram*.

Dalam *spectrogram* terdapat hubungan yang sangat erat antara waktu dan frekuensi. Hubungan antara frekuensi dan waktu adalah hubungan berbanding terbalik. Bila resolusi waktu yang digunakan tinggi, maka resolusi frekuensi yang dihasilkan akan semakin rendah.

2.3.7. Mel Frequency Wrapping

Mel Frequency Wrapping umumnya dilakukan dengan menggunakan *Filterbank*. *Filterbank* adalah salah satu dari bentuk filter yang dilakukan dengan tujuan untuk mengetahui ukuran energi dari *frequency band* tertentu dalam sinyal suara. *Filterbank* dapat diterapkan baik dalam domain waktu maupun pada domain frekuensi, tetapi untuk keperluan MFCC, *filterbank* harus diterapkan dalam domain frekuensi.



Gambar 2.7 Magnitude dari Rectangulare dan Tringular Filterbank

Sumber: Putra, (2009)

Filterbank menggunakan representasi konvolusi dalam melakukan filter terhadap sinyal. Konvolusi dapat dilakukan dengan melakukan multiplikasi antara spektrum sinyal dengan koefisien filterbank. Berikut ini adalah persamaan yang digunakan dalam filterbanks. (Putra, 2009:40):

$$Y[t] = \sum_{j=1}^N S[j]H_i [j] \quad (2-6)$$

Dimana:

N = jumlah magnitude spectrum.

S[j] = magnitude spectrum pada frekuensi j.

$H_i[j]$ = koefisien filterbank pada frekuensi j ($1 \leq i \leq M$).

M = jumlah channel dalam filterbank.

Presepsi manusia terhadap frekuensi dari sinyal suara tidak mengikuti linier scale. Frekuensi yang sebenarnya (dalam Hz) dalam sebuah sinyal akan diukur manusia secara subyektif dengan menggunakan mel scale. Mel frequency scale adalah linier frekuensi scale pada frekuensi dibawah 1000 Hz dan merupakan logarithmic scale pada frekuensi diatas 1000 Hz.

2.3.8. Discrete Cosine Transform (DCT)

DCT merupakan langkah terakhir dari proses utama MFCC *feature extraction*. Konsep dasar dari DCT adalah mendekorelasikan *mel spectrum* sehingga menghasilkan representasi yang baik dari properti spektral lokal. Pada dasarnya konsep DCT sama dengan *inverse fourier transform*. Namun hasil dari DCT mendekati PCA (*principle component analysis*). PCA adalah metode statistik klasik yang digunakan secara luas dalam analisa data dan kompresi. Hal inilah yang menyebabkan seringkali DCT menggantikan *inverse fourier transform* dalam proses MFCC *feature extraction*. Berikut adalah persamaan yang digunakan untuk menghitung DCT. (Putra, 2009:41):

$$C_n = \sum_{k=1}^K (\log S_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right]; n = 1, 2, \dots, K \quad (2-7)$$

dimana:

S_k = keluar dari proses *filterbank* pada *index k*.

K = jumlah koefisien yang diharapkan.

Koefisien ke nol dari DCT pada umumnya akan dihilangkan, walaupun sebenarnya mengindikasikan energi dari *frame* sinyal tersebut. Hal dilakukan karena, berdasarkan penelitian-penelitian yang pernah dilakukan, koefisien ke nol ini tidak *reliable* terhadap *speaker recognition*.

2.3.9. Cepstral Liftering

Hasil dari proses utama MFCC *feature extraction* memiliki beberapa kelemahan. *Low order* dari *cepstral coefficients* sangat sensitif terhadap *spectral slope*, sedangkan bagian *high order*-nya sangat sensitif terhadap *noise*. Oleh karena itu, *cepstral liftering* menjadi salah satu standar teknik yang diterapkan untuk meminimalisasi sensitifitas tersebut. *Cepstral liftering* dapat dilakukan dengan mengimplementasikan fungsi *window* terhadap *cepstral feature*. *Cepstral liftering* menghaluskan spektrum hasil dari *main processor* sehingga dapat digunakan lebih baik untuk *pattern matching*.

2.4 Modul *EasyVR*

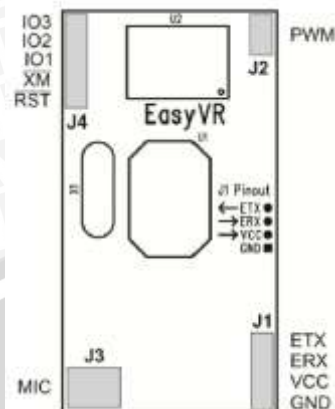
EasyVR merupakan modul *voice recognition* multi-fungsi. Dapat digunakan pada banyak aplikasi pengontrolan yang membutuhkan pendeteksian bukan hanya suara melainkan percakapan. *EasyVR* merupakan generasi penerus setelah kesuksesan generasi pertamanya di pasaran yaitu *VRBot*. Modul ini dapat digunakan atau dihubungkan dengan board mikrokontroler *Arduino*. Sangat cocok digunakan untuk beragam aplikasi, seperti *home automation* (dimana kita dapat mengontrol nyala lampu, kunci pintu, televisi, atau perangkat lainnya) atau sebagai modul pelengkap sensor pendengaran robot yang dibuat sebagaimana robot-robot canggih yang dijual di pasaran yang harganya luar biasa mahal.

Secara umum, fitur dari *EasyVR* adalah sebagai berikut:

- Mendukung beberapa bahasa, yaitu: *English(US), Italian, German, French, Spanish, Japanese*.
- Mendukung hingga 32 *custom Speaker Dependent (SD) trigger* atau perintah, bahkan dapat digunakan pada bahasa apapun.
- GUI yang mudah digunakan.
- Mudah diaplikasikan dan didukung oleh dokumentasi yang sederhana.
- 3 x GPIO (IO1, IO2, IO3) dapat dikontrol dengan perintah protokol baru.
- PWM audio output mendukung *speaker* 8 ohm.
- *Sound playback*.

Modul *EasyVR* dapat digunakan dengan antarmuka UART yang didukung pada rentang tegangan 3.3V - 5V, seperti PIC dan board *Arduino*. Beberapa contoh aplikasi termasuk otomatisasi rumah, seperti suara yang mengendalikan *switch* pada lampu, kunci, atau penambahan "pendengaran" untuk robot yang saat ini sedang berkembang.

2.4.1. Fungsi Pin pada EasyVR



Gambar 2.8 Konfigurasi pin pada EasyVR.

Sumber : Tigal, (2012)

Konfigurasi pin pada EasyVR yang ditunjukkan dalam Gambar 2.10 dan fungsi dari masing-masing pin ditunjukkan dalam Tabel 2.1.

Tabel 2.1 Fungsi pin EasyV.

KONEKTO R	NOMO R	NAMA	TIP E	DESKRIPSI
J1	1	GND	-	Ground
	2	VCC	I	Sumber tegangan DC input
	3	ERX	I	Serial Data Receive
	4	ETX	O	Serial Data Tranceive
J2	1 – 2	PWM	O	Diferensial audio output (mampu speaker 8 ohm)
J3	1	MIC_RE T	-	Ground pada microphone
	2	MIC_IN	I	Sinyal input pada microphone
J4	1	RST	I	Reset (logika LOW)
	2	XM	I	Boot select



3	IO1	I/O	General purpose I/O (3.0 VDC TTL level)
4	IO2	I/O	General purpose I/O (3.0 VDC TTL level)
5	IO3	I/O	General purpose I/O (3.0 VDC TTL level)

Sumber: Tigel, (2012)

2.4.2. Karakteristik Elektrik *EasyVR*

Karakteristik elektrik *EasyVR* ditunjukkan dalam Tabel 2.2.

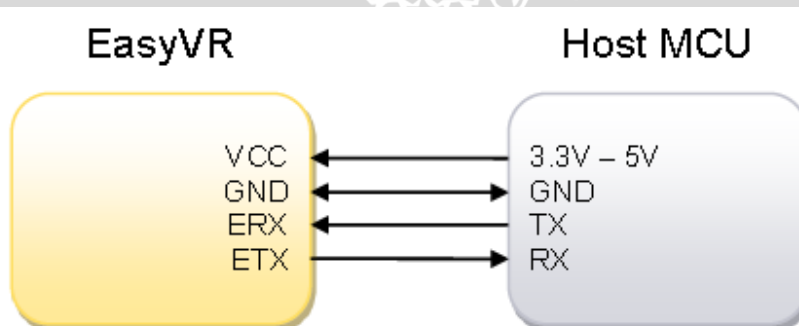
Tabel 2.2. Karakteristik elektrik *EasyVR*.

Symbol	Parameter	Min	Typ	Max	Unit
V _{IH}	Input High Voltage	2.4	3.0	3.3	V
V _{IL}	Input Low Voltage	-0.1	0.0	0.75	V
I _{IL}	Input Leakage Current (0 < V _{IO} < 3V, Hi-Z Input)		<1	10	μA
R _{PU}	Pull-up Resistance	Strong	10		kΩ
		Weak	200		kΩ
V _{OH}	Output High Voltage (I _{OH} = -5 mA)	2.4			V
V _{OL}	Output Low Voltage (I _{OL} = 8 mA)			0.6	V

Sumber : Tigel, (2012)

2.4.3. Communication Protocol

Komunikasi dengan modul *EasyVR* menggunakan antarmuka UART dengan standar kompatibel 3,3-5V TTL / CMOS sesuai dengan tegangan VCC.



Gambar 2.9 Protokol komunikasi pada modul *EasyVR*

Sumber: Tigel, (2012)

Konfigurasi awal di *power on* adalah 9600 *baud*, 8 bit data, *no parity*, 1 bit *stop*. *Baud rate* dapat diubah untuk beroperasi dalam kisaran 9600-115200 *baud*.

Protokol komunikasi hanya menggunakan karakter ASCII yang dapat dicetak, yang dapat dibagi dalam dua kelompok utama:

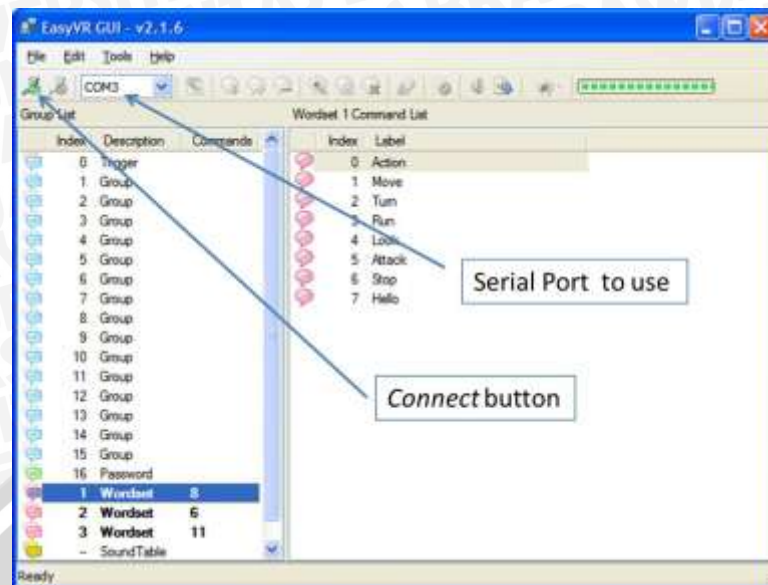
- *Command* dan status karakter, masing-masing pada TX dan RX, dipilih di antara huruf kecil.
- Argumen perintah atau rincian status, pada TX dan RX, mencakup berbagai huruf kapital.

Setiap perintah yang dikirim pada TX, dengan nol atau lebih argumen tambahan, menerima jawaban pada RX dalam bentuk status byte yang diikuti oleh nol atau lebih argumen. Terdapat penundaan minimum sebelum setiap byte dikirim keluar dari modul *EasyVR* ke RX yang awalnya ditetapkan 20 ms dan dapat dipilih kemudian dalam rentang 0 - 9 ms, 10-90 ms, 100 ms - 1 s. Karena pada antarmuka serial *EasyVR* juga berbasis *software*, penundaan yang sangat pendek mungkin diperlukan sebelum mengirimkan karakter ke modul, terutama jika *host* bekerja sangat cepat, sehingga memungkinkan *EasyVR* untuk kembali mendengarkan karakter baru.

2.4.4. *EasyVR Commander*

EasyVR commander adalah *software* yang digunakan untuk mengkonfigurasi modul *EasyVR* yang terhubung ke PC melalui adaptor atau dengan menggunakan mikrokontroler yang menyediakan program "*bridge*".

Pengguna dapat menentukan kelompok perintah atau *password* dan menghasilkan kode *template* untuk menangani mereka. Hal ini diperlukan untuk mengedit kode yang dihasilkan untuk mengimplementasikan aplikasi logika. *Template* berisi semua fungsi atau subrutin untuk menangani tugas-tugas pengenalan suara.



Gambar 2.10 Tampilan pada *EasyVR Commander*.

Sumber : Tigal, (2012)

Terdapat empat jenis perintah dalam *EasyVR Commander*, yaitu:

- *Trigger* : kelompok khusus di mana Anda memiliki built-in memicu kata SI "Robot" dan Anda dapat menambahkan satu user-defined memicu kata SD. Kata-kata pemicu digunakan untuk memulai proses pengenalan.
- *Group* : untuk menambhakan perintah SD user-defined.
- *Password* : kelompok khusus untuk "password vokal" (sampai lima), menggunakan *Speaker Verifikasi* (SV) teknologi.
- *Wordset* : built-in set perintah SI (misalnya pada Gambar 1 di atas, *Wordset* 1 dipilih).

2.5. Sensor *Microphone*

Microphone adalah sebuah transduser sinyal akustik ke sinyal listrik atau sensor yang mengubah sinyal suara menjadi sinyal listrik. *Microphone* dapat digunakan dalam berbagai aplikasi seperti telepon, *tape recorder*, karaoke, alat bantu dengar, teknik *live* audio dan rekaman, radio FRS, megafon, penyiaran televisi dan komputer untuk merekam suara, pengenalan suara, dan untuk tujuan non-akustik seperti pemeriksaan ultrasonik atau sensor ketukan.

2.5.1. Jenis-jenis *Microphone* berdasarkan Kepekaannya

Microphone memiliki berbagai jenis. Berikut ini adalah jenis-jenis *microphone* berdasarkan kepekaannya:

- 1) *Omnidirectional microphone* : jenis *microphone* yang dapat menerima suara dari segala arah.
- 2) *Bidirectional microphone* : jenis *microphone* yang mampu menerima suara dari depan dan belakang, sedangkan jika dari samping tidak peka.
- 3) *Unidirectional microphone* : jenis *microphone* yang hanya mampu menerima suara dari satu arah saja.

2.5.2 Sifat-sifat *Microphone*

Microphone terbagi dua menurut sifatnya, yaitu:

1) *Microphone Dynamic*

Microphone dynamic adalah suatu transducer yang menghasilkan sinyal listrik, akibat perubahan garis gaya magnet. Garis gaya magnet berubah-ubah karena tekanan yang berubah-ubah pada membran. Membran bergerak karena perubahan getaran udara yang mengenai membran.

2) *Microphone Condensor*

Microphone Condensor adalah suatu transducer yang mampu menghasilkan sinyal listrik. Sinyal listrik yang dihasilkan adalah karena perubahan nilai kapasitansi pada *microphone* tersebut. *Microphone condensar* disuplai dengan batu baterai sebesar 1,5 volt atau lebih tergantung dari konstruksinya.

2.6. *Arduino UNO*

Arduino UNO adalah *board* berbasis mikrokontroler pada ATmega328. *Board* ini memiliki 14 digital *input / output* pin (dimana 6 pin dapat digunakan sebagai *output PWM*), 6 *input analog*, 16 MHz *osilator* kristal, koneksi USB, jack listrik tombol reset. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber

tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya.



Gambar 2.11 Board *Arduino UNO* dan kabel USB.

Sumber: Arduino, (2009)

Tabel 2.3 Karakteristik *Arduino Uno*

Mikrokontroler	ATMega328
Operasi Voltage	5 V
Input Voltage	7 – 12 V (rekomendasi)
Input Voltage	6 – 20 V (limit)
I/O	14 pin (6 pin untuk PWM)
Arus	50 mA
Flash Memory	32 KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 MHz

Sumber: Arduino, (2011)

2.6.1. Catu Daya

Arduino UNO dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (non-USB) daya dapat datang baik dari AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkannya *plug* pusat-positif 2.1mm ke dalam *board* colokan listrik. *Lead* dari baterai dapat dimasukkan ke dalam *header* pin Gnd dan Vin dari konektor *Power*. *Board* dapat beroperasi pada pasokan daya dari 6 - 20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun, pin 5V dapat

menyuplai kurang dari 5 volt dan *board* mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Rentang yang dianjurkan adalah 7 - 12 volt.

Pin catu daya adalah sebagai berikut:

- VIN. Tegangan input ke *board* Arduino ketika menggunakan sumber daya eksternal (sebagai lawan dari 5 volt dari koneksi USB atau sumber daya lainnya diatur). Anda dapat menyediakan tegangan melalui pin ini, atau, jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
- 5V. Catu daya diatur digunakan untuk daya mikrokontroler dan komponen lainnya di *board*. Hal ini dapat terjadi baik dari VIN melalui regulator *onboard*,
- atau diberikan oleh USB .
- 3,3 volt pasokan yang dihasilkan oleh regulator on-board. Menarik arus maksimum adalah 50 mA.
- GND

2.6.2. Memory

ATmega328 ini memiliki 32 KB dengan 0,5 KB digunakan untuk *loading file*. Ia juga memiliki 2 KB dari SRAM dan 1 KB dari EEPROM.

2.6.3. Input & Output

Masing-masing dari 14 pin digital pada *Arduino UNO* dapat digunakan sebagai input atau output, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up* internal dari 20-50 K. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 *USB-to-Serial* TTL.

- Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat *attachInterrupt ()* fungsi untuk rincian.
- PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit output PWM dengan *analogWrite ()* fungsi.
- SPI: 10 (SS), 11 (mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI, LED menyala, ketika pin adalah RENDAH, itu off.
- *UNO* memiliki 6 input analog, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara *default* sistem mengukur dari tanah sampai 5 volt.
- TWI: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- Aref. Referensi tegangan untuk input analog. Digunakan dengan *analogReference ()*.
- *Reset*.

2.6.4. Komunikasi

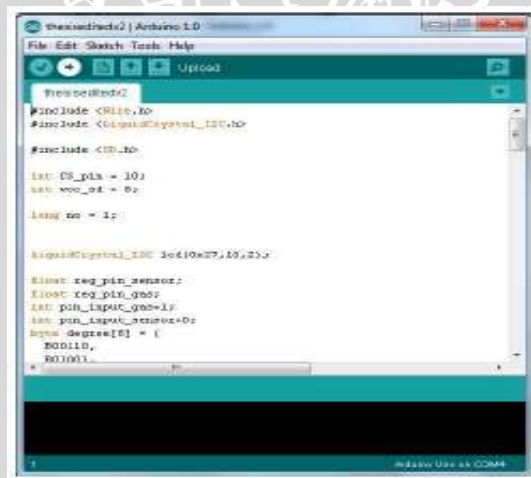
Arduino UNO memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, *Arduino* lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran *board* ini komunikasi serial melalui USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. *Firmware Arduino* menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Inf diperlukan. Perangkat lunak *Arduino* termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* *Arduino*. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi *inteface* pada sistem.

2.6.5. Programming

Arduino UNO dapat diprogram dengan perangkat lunak *Arduino*. Pilih *Arduino UNO* dari *Tool* lalu sesuaikan dengan mikrokontroler yang digunakan. ATmega328 pada *Arduino UNO* memiliki *bootloader* yang memungkinkan Anda untuk meng-upload program baru untuk itu tanpa menggunakan *programmer hardware* eksternal. Ini berkomunikasi menggunakan protokol dari bahasa C. Sistem dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau *programmer DFU* (Mac OS X dan Linux) untuk memuat *firmware* baru. Atau Anda dapat menggunakan *header ISP* dengan *programmer* eksternal.

2.6.6. Perangkat Lunak (*Arduino IDE*)

Lingkungan *open-source Arduino* memudahkan untuk menulis kode dan meng-upload ke *board Arduino*. Ini berjalan pada Windows, Mac OS X, dan Linux. Berdasarkan Pengolahan, *avr-gcc*, dan perangkat lunak sumber terbuka lainnya.



Gambar 2.12 Tampilan *framework Arduino*.
Sumber: Arduino, (2009)

2.6.7 Otomatis Software Reset

Tombol *reset Arduino UNO* dirancang untuk menjalankan program yang tersimpan didalam mikrokontroler dari awal. Tombol *reset* terhubung ke ATmega328 melalui kapasitor 100nf. Setelah tombol *reset* ditekan cukup lama

untuk me-reset chip, software IDE Arduino dapat juga berfungsi untuk meng-upload program dengan hanya menekan tombol upload di software IDE Arduino.

2.7. Relay

Relay adalah suatu piranti yang berfungsi untuk memutuskan atau menghubungkan satu atau lebih kontak elektronik. Relay berisi suatu komponen yang bila dimagnetisasi arus searah akan membangkitkan efek magnet yang dapat menghubungkan atau memutuskan kontak mekanik. Relay adalah sebuah saklar elektronis yang dapat dikendalikan dari rangkaian elektronik lainnya. Relay terdiri dari 3 bagian utama, yaitu:

1. koil : lilitan dari relay
2. common : bagian yang tersambung dengan NC(dalam keadaan normal)
3. kontak : terdiri dari NC (*Normaly close*) dan NO (*Normaly Open*)

Relay dapat bekerja karena adanya medan magnet yang digunakan untuk menggerakkan saklar. Saat kumparan diberikan tegangan sebesar tegangan kerja relay, maka akan timbul medan magnet pada kumparan karena adanya arus yang mengalir pada lilitan kawat. Kumparan yang bersifat sebagai elektromagnet ini kemudian akan menarik saklar dari kontak NC ke kontak NO. Jika tegangan pada kumparan dimatikan maka medan magnet pada kumparan akan hilang sehingga pegas akan menarik saklar ke kontak NC.

Fungsi Relay:

1. Digunakan untuk menjalankan fungsi logika (*Logic Function*)
2. Digunakan untuk memberikan fungsi penundaan waktu (*Time Delay Function*)
3. Digunakan untuk mengendalikan Sirkuit Tegangan Tinggi dengan bantuan dari Sinyal Tegangan Rendah.
4. Terdapat Relay yang berfungsi untuk melindungi Motor ataupun komponen lainnya dari kelebihan tegangan ataupun hubung singkat (*Short*)

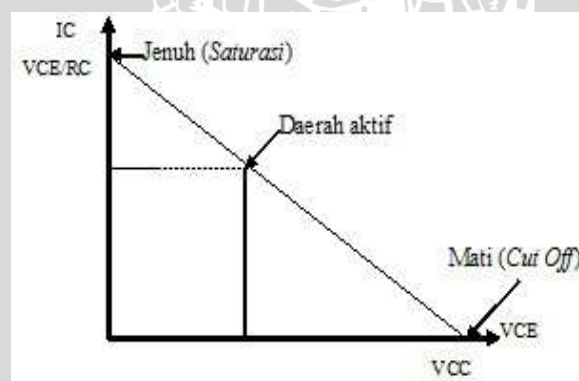
2.8 Transistor Sebagai Saklar

Salah satu fungsi transistor adalah sebagai saklar yaitu bila berada pada dua daerah kerjanya yaitu daerah jenuh (saturasi) dan daerah mati (*cut-off*). Transistor akan mengalami perubahan kondisi dari menyumbat ke jenuh dan

sebaliknya. Transistor dalam keadaan menyumbat dapat dianalogikan sebagai saklar dalam keadaan terbuka, sedangkan dalam keadaan jenuh seperti saklar yang menutup.

Daerah kerja transistor saat jenuh adalah keadaan dimana transistor mengalirkan arus secara maksimum dari kolektor ke emitor sehingga transistor tersebut seolah-olah *short* pada hubungan kolektor – emitor. Pada daerah ini transistor dikatakan menghantar maksimum (sambungan CE terhubung maksimum). Daerah aktif transistor pada daerah kerja ini transistor biasanya digunakan sebagai penguat sinyal. Transistor dikatakan bekerja pada daerah aktif karena transistor selalu mengalirkan arus dari kolektor ke emitor walaupun tidak dalam proses penguatan sinyal, hal ini ditujukan untuk menghasilkan sinyal keluaran yang tidak cacat.

Daerah aktif terletak antara daerah jenuh (saturasi) dan daerah mati (*Cut off*). Daerah *cut off* merupakan daerah kerja transistor dimana keadaan transistor menyumbat pada hubungan kolektor – emitor. Daerah *cut off* sering dinamakan sebagai daerah mati karena pada daerah kerja ini transistor tidak dapat mengalirkan arus dari kolektor ke emitor. Pada daerah *cut off* transistor dapat di analogikan sebagai saklar terbuka pada hubungan kolektor – emitor.

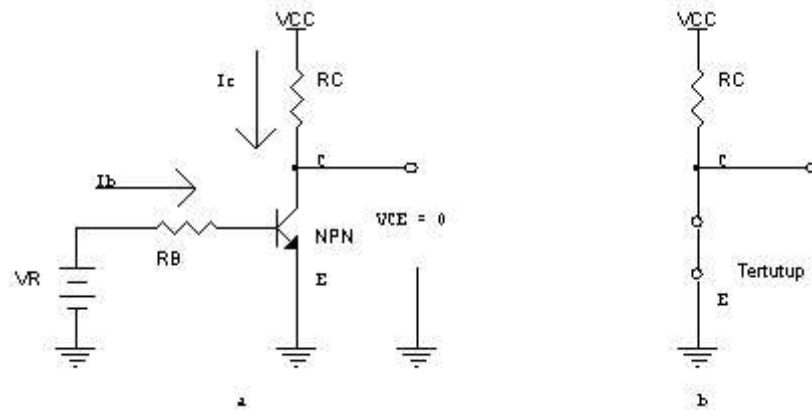


Gambar. 2.13. Grafik Kurva Karakteristik Transistor.

Sumber : Malvino, (1985)

Untuk membuat transistor menghantar, pada masukan basis perlu diberi tegangan. Besarnya tegangan harus lebih besar dari V_{be} (0,3 untuk *germanium* dan 0,7 untuk *silicon*). Dengan mengatur $I_b > I_c/\beta$ kondisi transistor akan menjadi jenuh seakan kolektor dan emitor *short circuit*. Arus mengalir dari kolektor ke emitor

tanpa hambatan dan $V_{ce} \approx 0$. Besar arus yang mengalir dari kolektor ke emitor sama dengan V_{cc}/R_c . Keadaan seperti ini menyerupai saklar dalam kondisi tertutup (ON).



Gambar. 2.14. Transistor Kondisi Jenuh (saklar posisi ON).

Sumber: Woolard, (2003)

Besarnya tegangan kolektor emitor V_{ce} suatu transistor pada konfigurasi diatas dapat diketahui sebagai berikut: (Malvino, 1985:121)

$$V_{ce} = V_{cc} - I_c \cdot R_c \tag{2-9}$$

Karena kondisi jenuh $V_{ce} = 0V$ (Transistor ideal) maka besarnya arus kolektor (I_c) adalah : (Malvino, 1985:122)

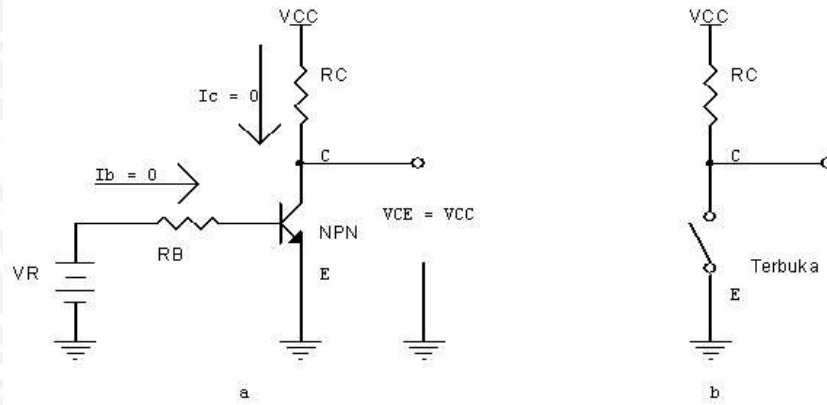
$$I_c = \frac{V_{cc}}{R_c} \tag{2-10}$$

Besarnya arus yang mengalir agar transistor menjadi jenuh (saturasi) adalah:

$$R_b = \frac{V_i - V_{be}}{I_b} \tag{2-11}$$

Sehingga besar arus basis I_b jenuh adalah:

$$I_b \geq \frac{I_c}{\beta} \tag{2-12}$$



Gambar. 2.15. Transistor Kondisi Mati (saklar posisi OFF).

Sumber : Woolard, (2003)

Dengan mengatur $I_b = 0$ atau tidak memberi tegangan pada bias basis atau basis diberi tegangan mundur terhadap emitor maka transistor akan dalam kondisi mati (*cut off*), sehingga tak ada arus mengalir dari kolektor ke emitor ($I_c \approx 0$) dan $V_{ce} \approx V_{cc}$. Keadaan ini menyerupai saklar pada kondisi terbuka seperti ditunjukkan pada gambar diatas. Besarnya tegangan antara kolektor dan emitor transistor pada kondisi mati atau *cut off* adalah: (Malvino, 1985:124)

$$V_{ce} = V_{cc} - I_c \cdot R_C \tag{2-13}$$

Karena kondisi mati $I_c = 0$ (transistor ideal) maka:

$$V_{ce} = V_{cc} \cdot R_c \tag{2-14}$$

$$V_{ce} = V_{cc} \tag{2-15}$$

Besar arus basis I_b adalah

$$I_b = \frac{I_c}{\beta} \tag{2-16}$$

$$I_b = 0 \tag{2-17}$$

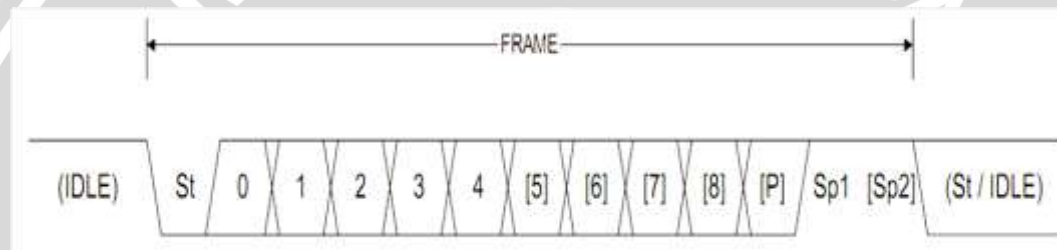
2.9. Komunikasi Serial

Pada prinsipnya, komunikasi serial ialah komunikasi dimana pengiriman data dilakukan per bit, sehingga lebih lambat dibandingkan komunikasi paralel seperti pada port printer yang mampu mengirim 8 bit sekaligus dalam sekali waktu. Devais pada komunikasi serial port dibagi menjadi 2 (dua) kelompok yaitu *Data Communication Equipment* (DCE) dan *Data Terminal Equipment* (DTE). Pengiriman data secara serial dapat berupa sinkron, yaitu pengiriman



clock dilakukan bersamaan dengan data, atau berupa asinkron, yaitu pengiriman clock tidak bersamaan dengan data namun secara dua tahap, saat data dikirim dan data diterima. Untuk istilah yang sering digunakan untuk mengirim dan menerima data secara asinkron biasa disebut *Universal Asynchronous Receiver Transmitter* (UART). Komunikasi data serial menggunakan UART sangat umum dan mudah penggunaannya, misalnya pada port serial PC. Pada UART jalur pengiriman dan penerimaan data serial dipisahkan.

Setiap pengiriman data pada UART menggunakan bit tanda *start* bit dan *stop* bit. Jalur data yang digunakan hanya satu untuk setiap pengiriman data. Data-data serial dikirim melewati jalur data satu persatu setiap satuan waktu. Format pengiriman data serial secara asinkron ditunjukkan dalam Gambar 2.17.

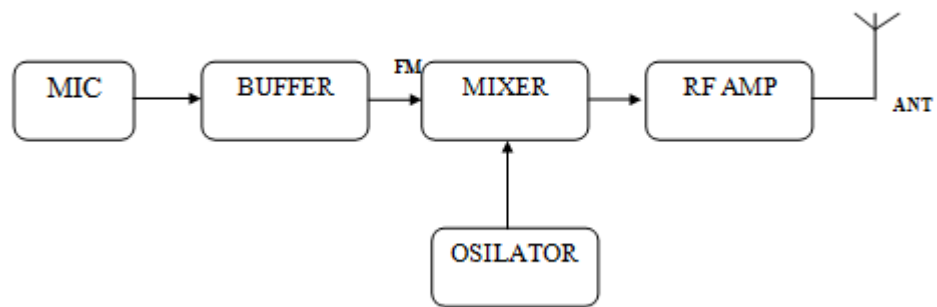


Gambar 2.16 Format frame data serial USART.

Sumber: Atmel, (2007)

2.10 Komunikasi *Wireless Microphone*

Wireless microphone merupakan *microphone* tanpa kabel. Melalui alat ini akan membuat praktis penggunaannya karena dapat menjangkau jarak yang relatif jauh tanpa memperhitungkan panjang kabel. Sinyal informasi dikirimkan dari *microphone* melalui antenna pemancar kemudian diterima oleh antena penerima dan disalurkan ke pengeras suara melalui penerima FM (Frekuensi Modulasi). *Wireless microphone* FM merupakan pemancar FM yang berfungsi untuk mengubah satu atau lebih sinyal input yang berupa *Audio Frequency* (AF) menjadi gelombang termodulasi dalam sinyal *Radio Frequency* (RF) yang kemudian output daya diumpankan ke sistem antenna untuk dipancarkan.



Gambar 2.17 Rangkaian pemancar FM

Sumber: Warren, (2000)

Rangkaian pemancar FM ini bekerja pada range frekuensi FM broadcast 88MHz – 108 MHz. Memiliki jarak pancar hingga jarak 20 – 30 meter. Dibutuhkan sumber tegangan 9 Volt untuk mengoperasikan pemancar ini.



BAB III

METODOLOGI PENELITIAN

Penyusunan penelitian ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasi sistem agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Langkah-langkah yang perlu dilakukan untuk merealisasikan sistem yang dirancang adalah penentuan spesifikasi alat, studi literatur, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan.

3.1 Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut:

- 1) Menggunakan *Arduino UNO* sebagai mikrokontroler utama berfungsi sebagai *host* mengatur komunikasi data dengan *EasyVR*
- 2) Pendeteksi suara menggunakan sensor suara *EasyVR* berfungsi sebagai masukan
- 3) Digunakan *micropone* yang berbasis *wireless* bekerja pada rentang modulasi frekuensi, berfungsi agar pengucapan perintah suara dapat dilakukan dengan rentang jarak yang lebih jauh tanpa menggunakan kabel.
- 4) *Relay* yang berfungsi sebagai saklar dalam hubungan dengan peralatan listrik sebagai penyambung dan pemutusan tegangan listrik.
- 5) Penelitian dilakukan untuk 3 beban yang berbeda yaitu lampu, kipas angin dan televisi.

3.2 Studi Literatur

Studi literatur dilakukan untuk mempelajari teori penunjang sistem yang dibutuhkan dalam perencanaan dan pembuatan alat. Teori yang diperlukan antara lain berkaitan dengan *Relay*, sistem *voice recognition*, modul *EasyVR*, modul *wireless*, Catu daya, dan *Arduino UNO*.

3.3 Perancangan *Voice Recognition* Menggunakan *Easy VR*

Bertujuan untuk mengambil sample suara yang akan disimpan ke dalam modul *EasyVR*. Pengambilan sampling suara dilakukan melalui PC dengan *software* bawaan dari *EasyVR* yaitu *EasyVR commander*.

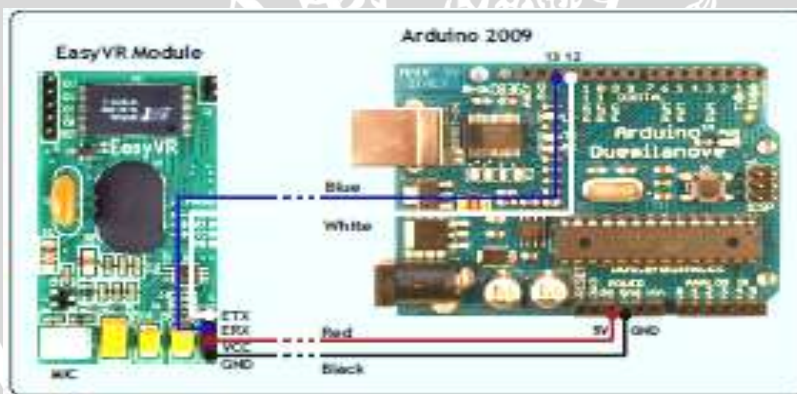
3.3.1 Perancangan Sistem Komunikasi *EasyVR* ke Komputer

Proses komunikasi antara *EasyVR* dengan komputer berfungsi untuk sampling data berupa suara melalui *software EasyVR Commander*. Komunikasi antara *EasyVR* menggunakan komunikasi serial USB to TTL.



Gambar 3.1. Proses Komunikasi *EasyVR* dengan komputer.

3.3.2 Perancangan Sistem Komunikasi *EasyVR* ke *Arduino Uno*

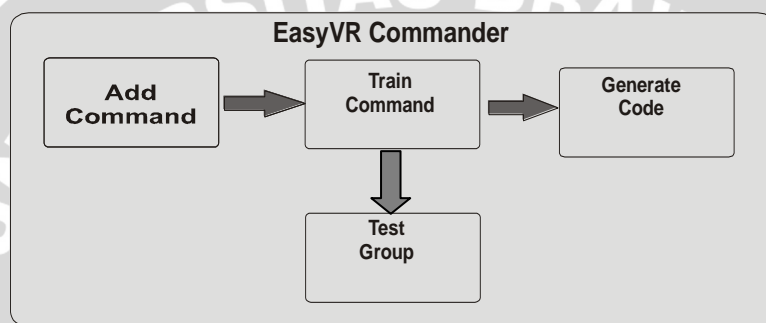


Gambar 3.2. Proses komunikasi *EasyVR* dengan *Arduino UNO*.

Komunikasi antara *EasyVR* dengan *Arduino UNO* menggunakan *bridge Mode* atau secara serial. Pin 13 pada *Arduino UNO* difungsikan sebagai Tx yang akan dihubungkan dengan Rx dari *EasyVR*. Sedangkan pin 12 pada *Arduino UNO* difungsikan sebagai Rx yang akan dihubungkan dengan Tx pada *EasyVR*.

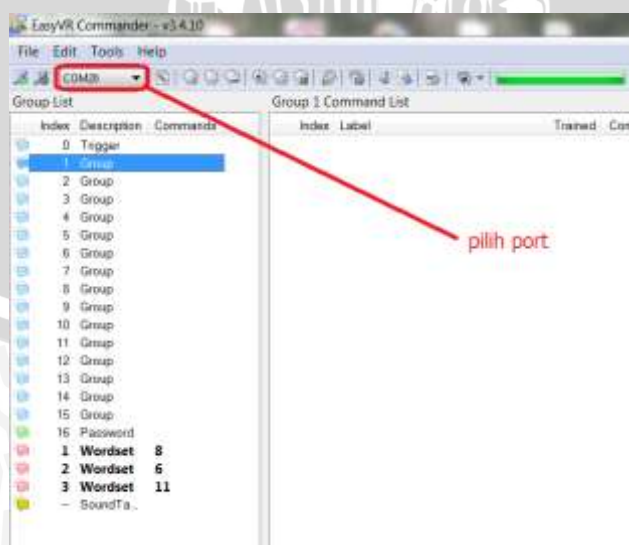
Hasil sampling data yang tersimpan dalam *EasyVR* tidak dikirimkan ke mikrokontroler yang terdapat pada *Arduino UNO*, namun tetap tersimpan di dalam *EasyVR*. Prinsip kerjanya adalah pada saat input berupa suara sesuai dengan hasil sampling, *EasyVR* akan memberi perintah kepada *Arduino UNO* berupa data serial 8 bit dengan kecepatan 9600 bps. Jika *Arduino* menerima data karakter ASCII “A” (dalam bentuk desimal 65 hex atau hex 41) dari *EasyVR*, maka akan muncul data berupa 10000010.

3.3.3. Perancangan Pengambilan *Sample* Suara Menggunakan *EasyVR Commander*



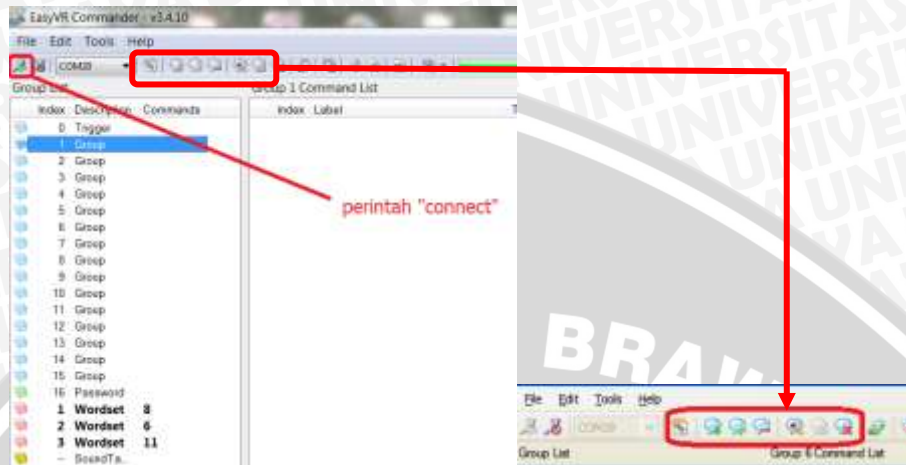
Gambar 3.3. Blok diagram sistem *voice reognition* melalui *EasyVR*.

Sebelum modul *EasyVR* dihubungkan ke PC dengan menggunakan antarmuka serial berupa *USB to TTL*, *software EasyVR Commander* harus diaktifkan terlebih dahulu. Kemudian terdapat pilihan *port* USB yang terhubung oleh modul *EasyVR* yang ditunjukkan dalam Gambar 3.4.



Gambar 3.4. Tampilan pada *software EasyVR Commander*

Kemudian tekan perintah *connect* agar *EasyVR* dapat terhubung dengan software tersebut dan untuk mengaktifkan berbagai perintah yang terdapat dalam software tersebut seperti yang ditunjukkan dalam Gambar 3.5.



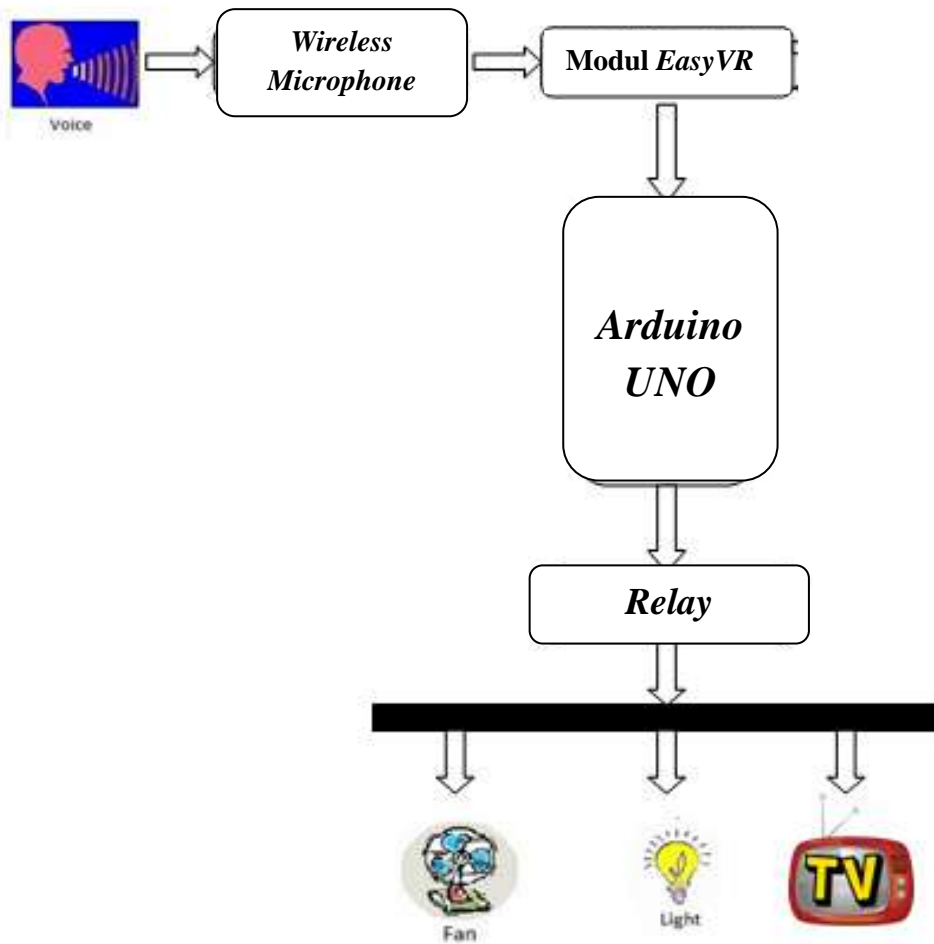
Gambar 3.5. Tampilan pada software *EasyVR Commander*

Terdapat beberapa command yang dapat digunakan untuk memperoleh *sampling* suara. Langkah pertama untuk mendapatkan *sampling* suara adalah dengan menambahkan *command* ke dalam grup melalui perintah “*add command*” yang terdapat dalam software. Kemudian memilih perintah “*train command*” untuk memasukkan *sampling* suara yang akan digunakan. *Command* yang telah ditambahkan akan secara langsung tersimpan dalam *database* pada modul *EasyVR*. *Generate code* berguna untuk menunjukkan *code* dalam bahasa C untuk mempermudah pemrograman pada perancangan software.

3.4 Perancangan dan Penyusunan Perangkat Lunak

Penyusunan perangkat lunak digunakan untuk mengendalikan dan mengatur kerja dari alat ini. Desain dan parameter yang telah dirancang kemudian diterapkan kedalam program *Arduino* dengan menggunakan bahasa pemrograman C.

3.4.1 Perancangan Sistem Secara Keseluruhan



Gambar 3. 6. Diagram blok sistem secara keseluruhan.

Fungsi masing-masing bagian dalam diagram blok ini adalah sebagai berikut:

- 1) Sensor *microphone* berfungsi untuk mendeteksi suara.
- 2) *Wireless microphone* berfungsi menjangkau jarak yang relatif jauh tanpa memperhitungkan panjang kabel.
- 3) *EasyVR* berfungsi sebagai antarmuka yang berisi *database* berupa *sampling* suara yang telah dikonfigurasi dengan *EasyVR Commander Arduino UNO* dengan sensor *microphone*.

- 4) *Arduino UNO* yang berfungsi sebagai MK Utama untuk memanggil dan mengolah data-data berupa suara dari *EasyVR* yang kemudian akan diolah untuk proses dan sebagai penghubung dengan *Relay*.
- 5) *Relay* berfungsi sebagai saklar *ON OFF* yang dihubungkan dengan beban peralatan listrik.

Prinsip kerja dari perancangan adalah peralatan listrik dihubungkan dengan *stop* kontak yang terpasangan pada alat, terdapat 3 buah *stop* kontak. Setelah itu pemberian input dilakukan dengan menggunakan perintah suara. Data akan diakses oleh *EasyVR* yang akan melakukan pemilihan jenis suara dan akan diolah oleh mikrokontroler yang melakukan pemrosesan data untuk melaksanakan perintah yang diberikan.

3.5 Pengujian Alat

Pengujian alat dilakukan untuk memastikan bahwa sistem ini berjalan sesuai dengan yang direncanakan. Pengujian alat ini meliputi pengujian perangkat keras yang dilakukan baik per blok maupun keseluruhan sistem.

3.5.1 Pengujian Tiap Blok

Pengujian per blok dilakukan dengan tujuan untuk menyesuaikan nilai masukan dan nilai keluaran tiap-tiap blok sesuai dengan perancangan yang dilakukan sebelumnya. Meliputi blok *wireless microphone*, blok modul *EasyVR*, blok mikrokontroler utama, blok *relay* sebagai saklar pengendali.

3.5.2 Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama. Pengujian dilakukan terhadap parameter uji yaitu pemberian perintah dari orang yang berbeda, keberhasilan menerima perintah dari jarak jauh menggunakan *wireless microphone*, serta lama waktu respon peralatan listrik setelah dilakukan pemberian perintah melalui ucapan suara.

3.6 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang

direncanakan maka alat tersebut telah memenuhi harapan dan memerlukan pengembangan untuk penyempurnaannya.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

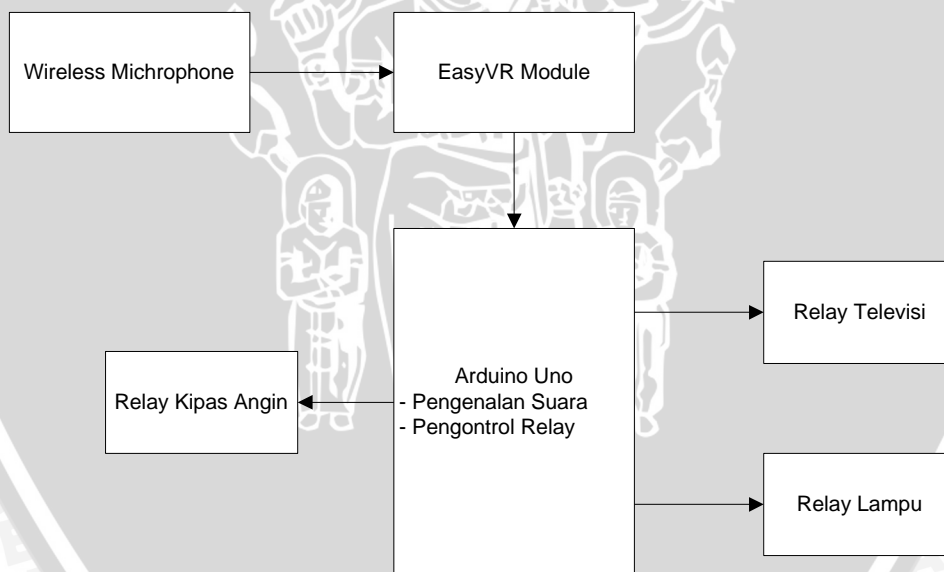
Perancangan alat ini dilakukan secara bertahap sehingga akan memudahkan dalam analisis pada setiap bloknnya maupun secara keseluruhan.

Perancangan ini terdiri dari:

- Perancangan sistem
- Perancangan perangkat keras
- Perancangan sistem voice recognition
- Perancangan perangkat lunak

4.1 Perancangan Sistem

Blok diagram keseluruhan sistem yang dirancang dibagi menjadi dua bagian, yaitu blok mikrokontroler utama, dan blok *EasyVR* ditunjukkan dalam Gambar 4.1.



Gambar 4.1. Diagram Blok Sistem

Fungsi masing-masing blok dalam Gambar 4.1 adalah sebagai berikut:

1). Blok *Wireless Microphone*

Terdapat *microphone* dan *antenna* sebagai pemancar, untuk menangkap sinyal suara, sehingga pengucapan dapat dilakukan dari

jarak jauh, bekerja pada Frekuensi modulasi, dengan rentang *broadcast* antara 88 Mhz – 108 Mhz.

2). Blok Modul *EasyVR*

Pengendalian dilakukan melalui suara yang dibantu dengan modul *EasyVR* yang berfungsi sebagai sensor suara. Data-data suara akan di *sampling* melalui komputer menggunakan program *EasyVR Commander*. Data-data yang sudah disampling akan dimasukkan kembali ke database *EasyVR*. Mikrokontroler utama berupa *Arduino UNO* bekerja mengatur dan mengolah data dari modul *EasyVR*.

3). Blok Mikrokontroler Utama

Mikrokontroler yang digunakan adalah *Arduino UNO* yang berfungsi untuk mengolah data dari modul *EasyVR* dan mengakses *relay* untuk mengendalikan peralatan listrik (lampu, kipas angin dan televisi).

4). Blok *Relay* sebagai saklar

Dilakukan percobaan terdapat 3 jenis peralatan listrik rumah tangga, yaitu televisi lampu dan kipas angin. *Relay* digunakan sebagai saklar terhadap 3 jenis peralatan listrik tersebut.

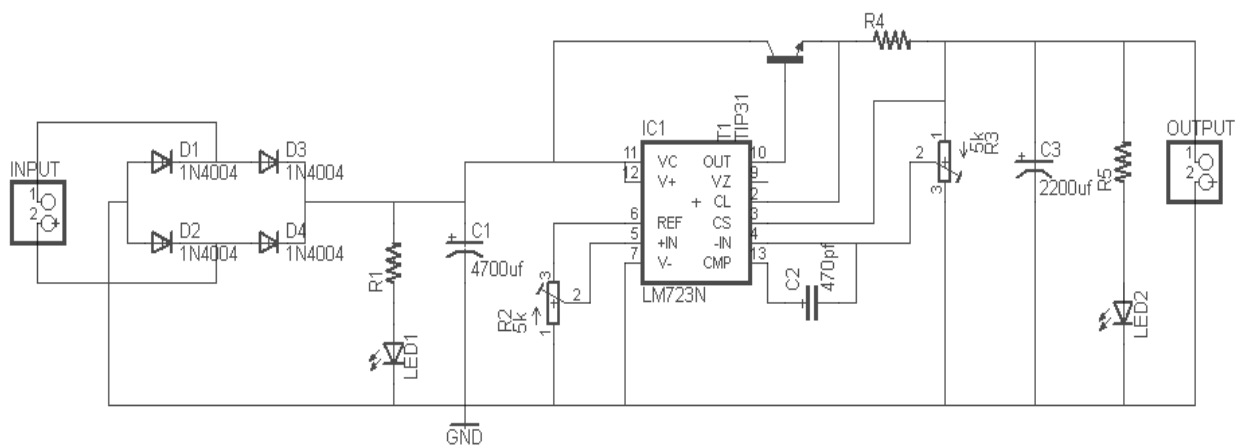
4.2 Perancangan Perangkat Keras

4.2.1 Perancangan Desain Sistem Elektronik

Diagram blok sistem elektronik, terdiri dari tiga bagian catu daya, perancangan relay mekanik, perancangan mikrokontroler utama dan perancangan sistem *voice recognition* menggunakan *EasyVR*. Pada bagian masukan berupa sebuah sensor *microphone* yang berfungsi untuk menerima masukan berupa suara manusia yang terhubung dengan *EasyVR* sebagai pengolah data sinyal suara. Pada bagian kendali menggunakan *ArduinoUNO* sebagai mikrokontroler utama dan pengontrolan saklar.

4.2.2. Perancangan Catu Daya Sistem

Arduino UNO dapat bekerja jika diberi catu tegangan antara 4,8 V hingga 5,5 V sesuai dengan *datasheet Arduino UNO* yang meliputi board *EasyVR*. Pada perancangan digunakan catu daya sebesar 5 V yang diperoleh dari IC regulator LM723 agar diperoleh tegangan keluaran sebesar 5 V. Skema rangkaian catu daya ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Rangkaian Catu Daya

Dioda *bridge* berfungsi sebagai *rectifier* atau penyearah arus dari AC ke DC. Dipilih kapasitor yang mempunyai tegangan kerja yang lebih besar dari tegangan 11 Volt, perhitungan didapat dari

$$V_R = \frac{2V_p}{\pi} \tag{4-1}$$

$$V_p = V_s \cdot \sqrt{2} \tag{4-2}$$

Diketahui tegangan V_s sebesar 12 Volt

$$\begin{aligned} V_p &= V_s \cdot \sqrt{2} \\ &= 12 \cdot \sqrt{2} = 16.9 \text{ Volt} \end{aligned}$$

$$V_R = \frac{2V_p}{\pi} = \frac{2 \times 16,9}{3.14} = 10.8 \text{ Volt}$$

Kapasitor yang digunakan harus lebih besar dari nilai V_R (tegangan rata-rata) 10.8 Volt dapat dapat dibulatkan menjadi 11 Volt, dibutuhkan tegangan kerja kapasitor diatas 11 Volt kapasitor yang tersedia dipasaran mempunyai tegangan 25 Volt jadi menggunakan kapasitor dengan besar tegangan 25 Volt dengan besar muatan 4400 μF .

Terdapat tegangan sebesar 11 Volt, jadi pada saat menghantar dioda-dioda ini akan terkena tegangan sekitar 11 Volt. Tetapi pada saat diode ini tidak menghantar, diode mempunyai 2 tegangan terbalik, berarti dibutuhkan dioda yang mempunyai tegangan terbalik V_{ref} sebesar 2×11 Volt butuh tegangan terbalik sebesar 22 Volt jenis pilihan diode jatuh pada 1N 4002 yang mempunyai tegangan terbalik sebesar 200 Volt tetapi hanya memiliki besar arus yang tidak sesuai dengan arus yang dibutuhkan, yaitu hanya sebesar 1 A.

Tegangan yang keluar dari dioda penyearah masih terdapat *ripple*, kemudian disupress (ditekan) oleh IC LM723 untuk menghilangkan tegangan *ripple*. LM723 hanya mempunyai arus keluaran sebesar 150mA sedangkan besar arus yang diperlukan sebesar 2A, maka digunakan transistor yang dapat berfungsi sebagai penguat arus maupun tegangan. Transistor yang digunakan harus memiliki I_C (Arus Kolektor) lebih besar dari 2 Ampere dan harus memiliki I_B (Arus Basis) yang kecil dan lebih kecil dari 150mA.

Daya disipasi yang terdapat pada transistor

$$\begin{aligned} P_{\text{disipasi}} &= I_C \cdot V_{\text{CE}} & (4-3) \\ &= 2 \text{ A} \cdot (10.8 \text{ Volt} - 5 \text{ volt}) \\ &= 11.6 \text{ Watt} \end{aligned}$$

Digunakan transistor yang mempunyai $P_{\text{disipasi}} >$ dari 11.6 Watt maka digunakan transistor berjenis TIP31.

Perhitungan Resistor

$$R = \frac{V}{I} \quad (4-4)$$

$$I = \frac{\varepsilon V}{\varepsilon R} = \frac{(\text{Tegangan rata-rata}) - \text{Tegangan LED}}{R} \quad (4-5)$$

$$= \frac{10.8 \text{ V} - 1.8 \text{ V}}{R}$$

Diketaui I_{led} sebesar 50mA didapat dari datasheet.

$$\text{Sehingga } R_1 = \frac{9 \text{ Volt}}{50 \text{ mA}} = 1800 \Omega$$

$$P_{\text{disipasi1}} = I^2 \cdot R \quad (4-6)$$

$$= (50 \times 10^{-3})^2 \cdot 1800$$

$$= 0.045 \text{ Watt}$$

$$R_4 = \frac{V}{I} = \frac{0.6 \text{ Volt (Tegangan Transistor)}}{2 \text{ A (arus keluaran yang kita inginkan)}}$$

$$= 0.3 \Omega$$

$$P_{\text{disipasi4}} = I^2 \cdot R$$

$$= (2)^2 \cdot 0.3 = 1.2 \text{ Watt}$$

$$R_5 = \frac{\varepsilon V}{I} = \frac{\text{Tegangan keluaran dari IC} - \text{Tegangan LED}}{I}$$

Tegangan keluaran dari IC didapat sebesar 5 Volt dan tegangan LED sebesar 1.8 Volt didapat dari *datasheet*, sedangkan arus yang dipakai adalah arus pada LED sebesar 50mA juga diketahui dari *datasheet*.

Sehingga

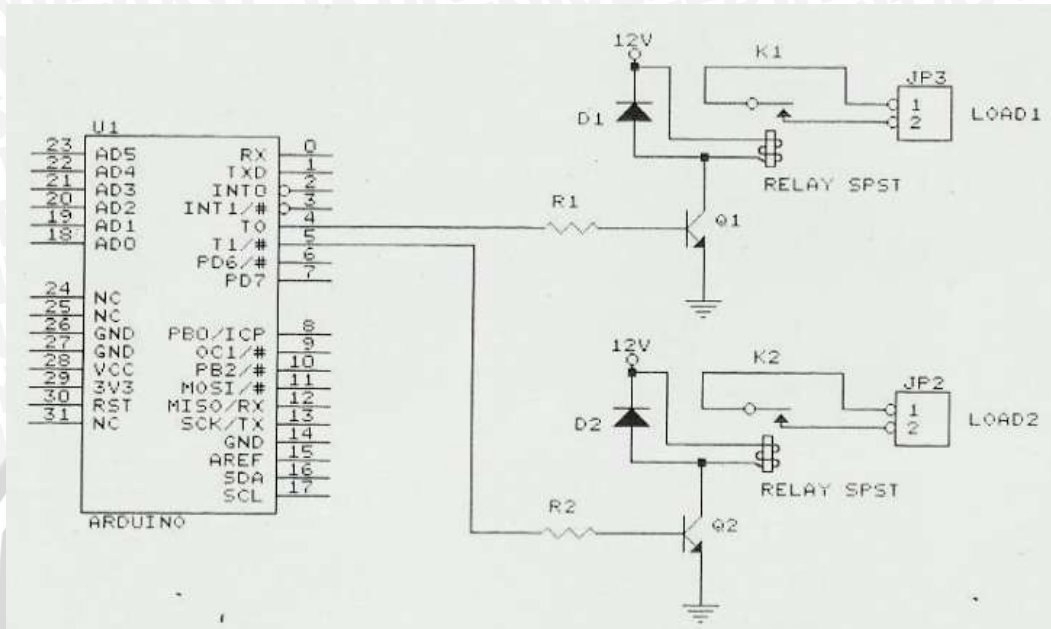
$$R_5 = \frac{5 \text{ V} - 1.8 \text{ V}}{50 \text{ mA}} = 64 \Omega$$

$$P_{\text{disipasi5}} = I^2 \cdot R$$

$$= (0.05)^2 \cdot 64 = 0.16 \text{ Watt}$$

4.2.3. Perancangan Rangkaian *Relay*

Relay digunakan sebagai saklar pengendali untuk peralatan listrik rumah.

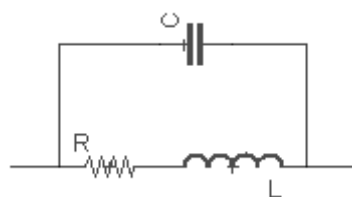


Gambar 4.3 Rangkaian *Relay* Mekanik

Dalam *relay* terdapat koil yaitu lilitan, lilitan-lilitan pada *relay* ini terdiri dari resistor, induktor dan kapasitor. Anggap saja koil pada *relay* hanya mempunyai R dan L karena kita menganalisa pada sumber DC. Pada analisa DC, $X_L = 0$. Kemudian lalu kita ukur menggunakan multimeter bahwa besar R pada koil sebesar 200Ω . Mengapa X_c diabaikan, dilihat dari persamaan 4.8 pada perhitungan X_c memiliki nilai frekuensi yang mana nilai frekuensinya sebesar 0 sedangkan *relay* dianalisa pada sumber DC, sehingga nilai X_c tak berhingga.

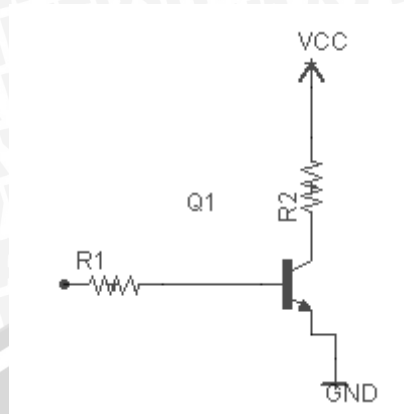
$$X_L = j\omega l \quad (4-7)$$

$$X_c = \frac{1}{2\pi f C} \quad (4-8)$$



Gambar 4.4 Rangkaian RLC

Menghitung arus koil pada *relay*



Gambar 4.5 Transistor pada *relay*

$$I_c \cdot R_{Dc \text{ koil}} \text{ maka } V_{CE} - V_{CC} = 0 \tag{4-9}$$

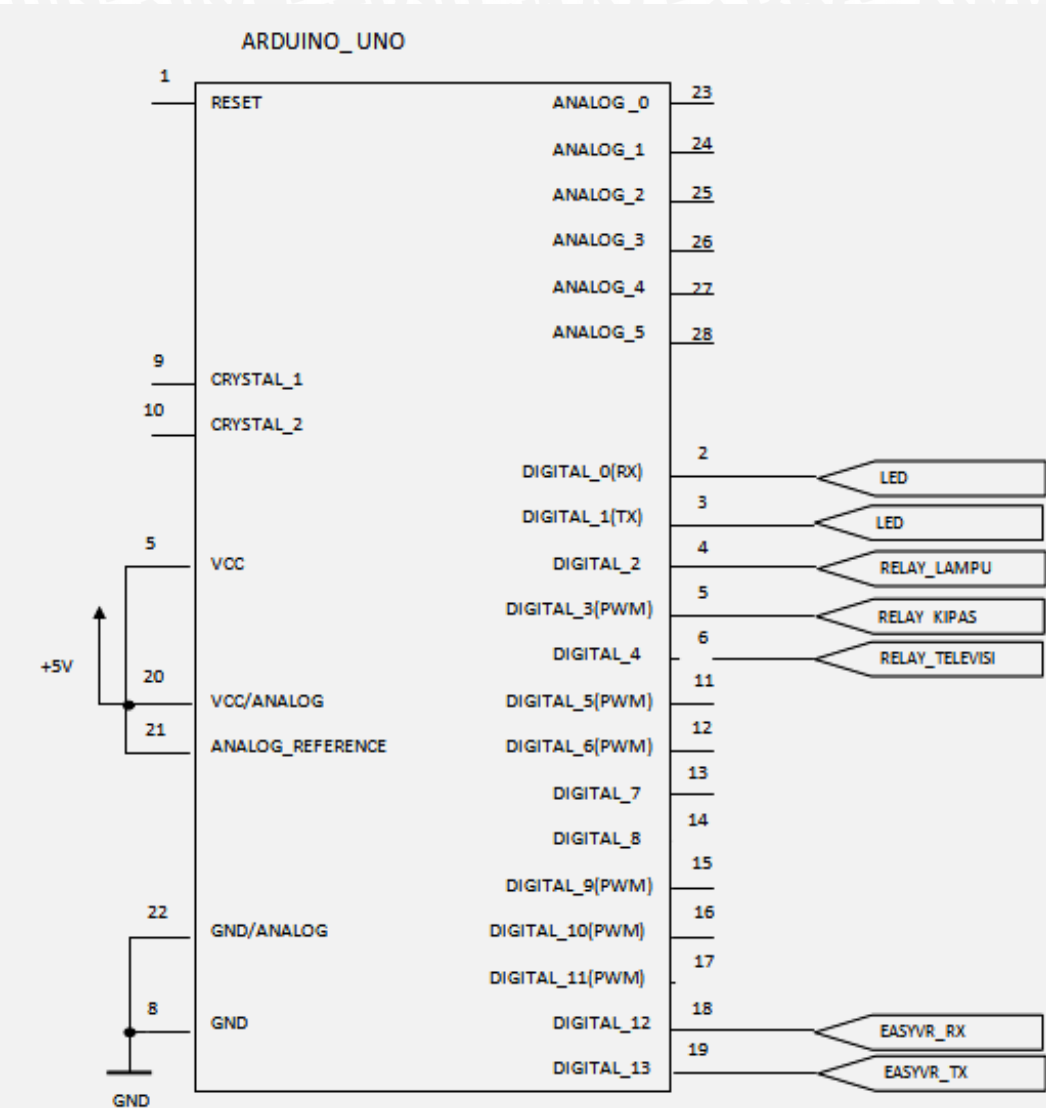
$$I_c = \frac{V_{cc} - V_{CE}}{R_{Dc \text{ koil}}} \tag{4-10}$$

$$= \frac{V_{CC}}{R_{Dc \text{ koil}}} = \frac{5V}{200\Omega}$$

$$= 25\text{mA}$$



4.2.4. Perancangan Rangkaian Mikrokontroler Pengatur Utama

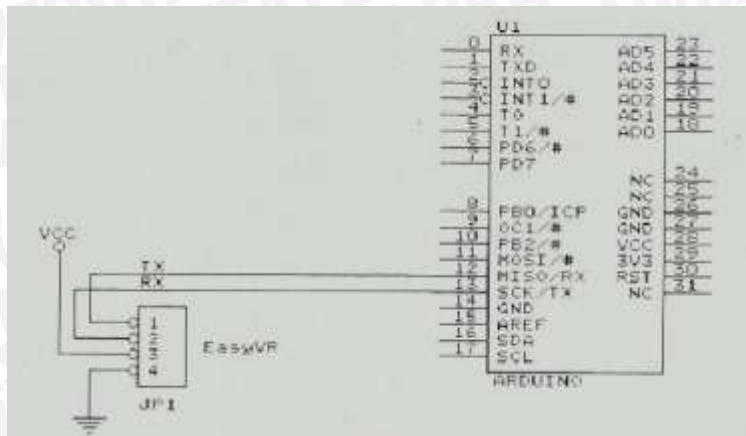


Gambar 4.6 Minimum Sistem *Arduino UNO*

Pada perancangan mikrokontroler pengatur utama menggunakan modul *Arduino UNO* sebagai pengolah utama dalam melakukan proses pengolahan data. *Arduino UNO* mempunyai 4 port, 32 jalur yang dapat diprogram menjadi masukan atau keluaran.

4.3 Hubungan *Arduino UNO* dengan *Easy VR*

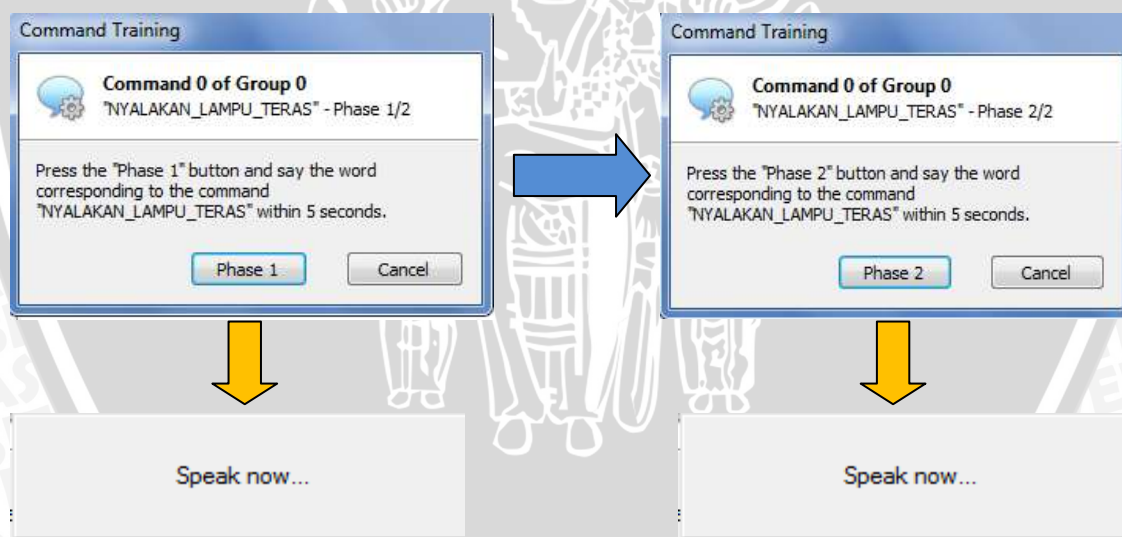
Komunikasi yang dilakukan antara *Arduino UNO* dengan *EasyVR* modul menggunakan TX dan RX. *Arduino UNO* berfungsi sebagai komunikasi data dengan *EasyVR*.



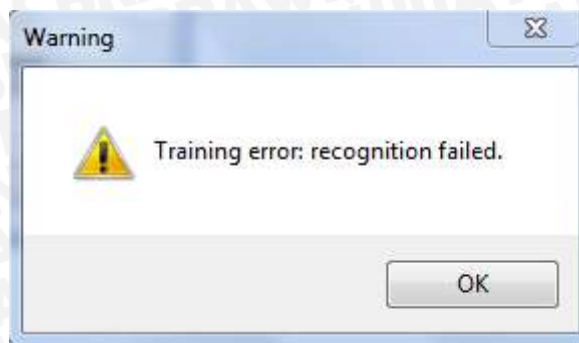
Gambar 4.7 Minimum Sistem EasyVR

4.3.1 Perancangan Sistem Pengenalan Suara Menggunakan EasyVR

Perancangan ini bertujuan untuk mengambil *sample* suara yang akan disimpan didalam modul EasyVR. Pengambilan *sample* suara dilakukan melalui PC dengan software bawaan dari EasyVR yaitu *EasyVR Commander*. *Sample* suara yang akan digunakan adalah sebanyak enam kata.

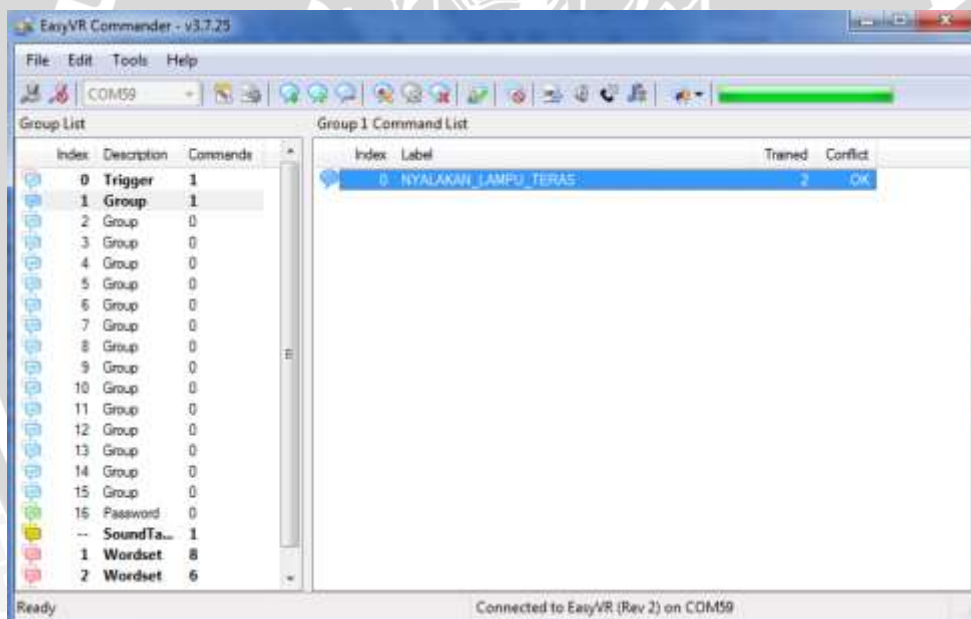


Gambar 4.8 Diagram blok pemberian *sample* suara



Gambar 4.9 Tampilan kegagalan dalam pengambilan *sample* suara

Pengambilan sampel suara dilakukan sebanyak dua kali dengan kondisi ideal atau tidak adanya *noise*, variasi kata dan intonasi disetiap pengucapan relatif sama menghasilkan kesuksesan tinggi dalam pengambilan *sample* dengan tidak adanya *error*. Kegagalan menerima variasi suara kedua dengan variasi pengucapan suara pertama akan menimbulkan kegagalan seperti ditunjukkan dalam gambar 4.9. Oleh karena itu, agar *EasyVR* dapat berfungsi dengan baik, dibutuhkan variasi suara yang relatif sama dengan *sample*.



Gambar 4.10 Sampling Suara Pada *EasyVR Commander*



Gambar 4.11 Proses Penyamplingan suara menggunakan *EasyVR*

4.4 Perancangan dan Pembuatan Perangkat Lunak (*Software*)

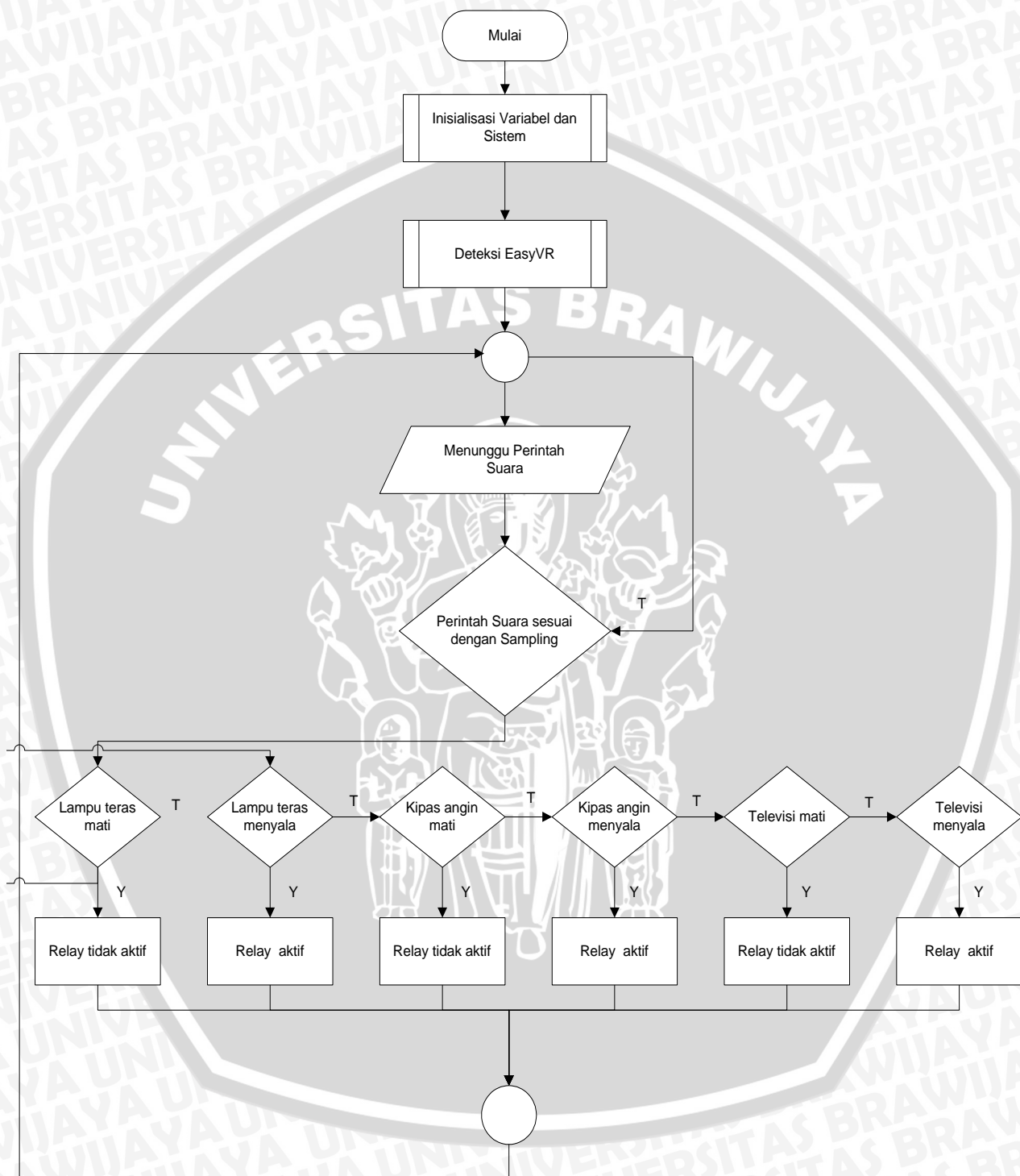
Setelah melalui proses perancangan dan pembuatan perangkat keras, selanjutnya akan dilakukan perancangan dan pembuatan perangkat lunak. Perangkat lunak berfungsi untuk memberikan instruksi kerja kepada perangkat keras tersebut.

4.4.1 Perancangan Susunan Perangkat Lunak

Tahapan proses yang terdapat pada sistem ini meliputi proses pengolahan data dari modul *EasyVR* ke *Arduino* dan proses pengontrolan *relay*. Semua proses tersebut dilakukan oleh perangkat lunak yang terdapat dalam mikrokontroler. Perangkat lunak ini tersusun dari instruksi-instruksi yang membentuk sebuah *listing* program atau *source code*.

Semua instruksi program disusun secara terstruktur dalam beberapa subrutin yang secara khusus menangani fungsi tertentu. *Software* mikrokontroler dibuat menggunakan program *Arduino*. Menggunakan bahasa pemrograman yaitu bahasa C.

4.4.2. Diagram Alir Keseluruhan Sistem

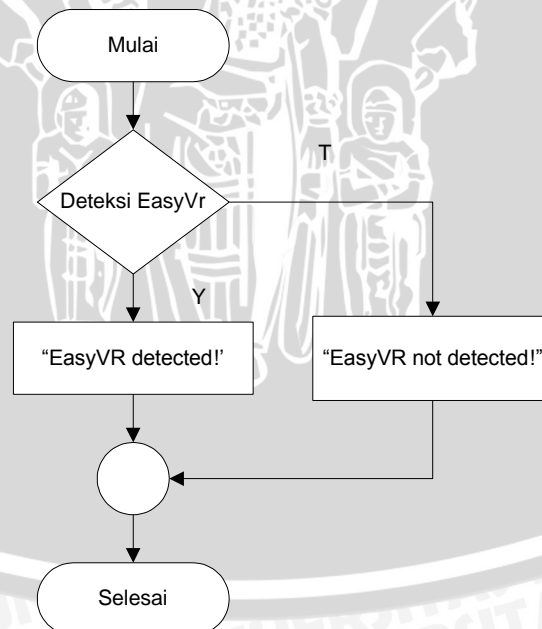


Gambar 4.12 Diagram alir keseluruhan sistem

Diagram alir keseluruhan sistem ditunjukkan dalam Gambar 4.12. Respon alat akan menginisialisasikan sistem yang didalamnya juga terdapat sistem pendeteksi keberadaan *EasyVR*. Kemudian menunggu perintah berupa suara. Setelah itu, data input berupa suara dibandingkan dengan data sampling yang ada pada *EasyVR*. Jika sesuai, *Board Arduino UNO* akan menerima perintah dari *board EasyVR* dan mengirimkan perintah kepada rangkaian *Relay*.

4.4.3 Diagram Alir Program Pendeteksi *EasyVR*

Pendeteksi *EasyVR* berfungsi untuk mengetahui apakah modul *EasyVR* sudah *connect* atau belum dengan *ArduinoUNO*. Keluaran dari program ini akan muncul pada serial monitor yang terdapat dalam *software Arduino*. Jika *EasyVR* terdeteksi, akan muncul kalimat “*EasyVR Detected*” pada tampilan awal serial monitor. Sebaliknya jika tidak terdeteksi, akan muncul kalimat “*EasyVR not Detected*”. Serial monitor selain berfungsi untuk mengetahui keberadaan *EasyVR*, juga untuk mengetahui pengiriman data suara apakah sesuai atau tidak. Gambar berikut menunjukkan diagram alir program pendeteksi *EasyVR*.



Gambar 4.13 Diagram alir program pendeteksi *EasyVR*

BAB V

PENGUJIAN DAN ANALISIS

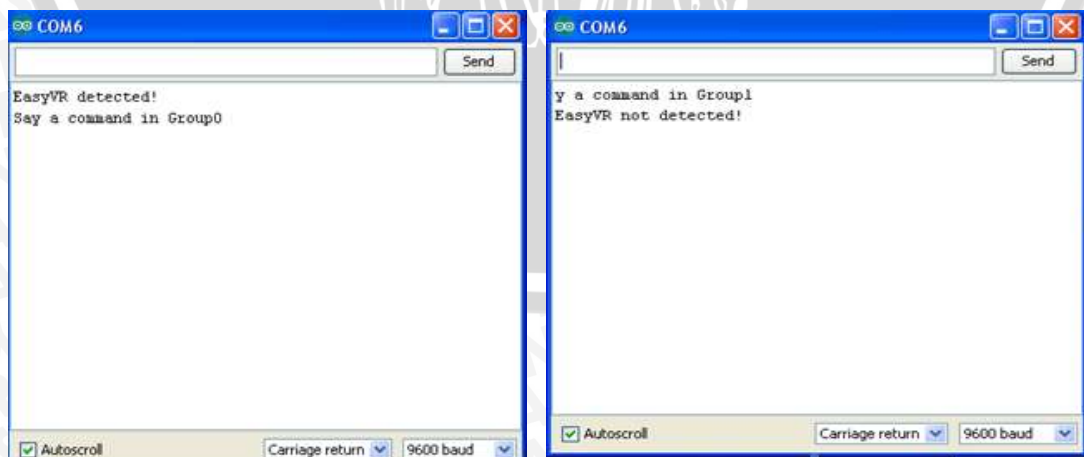
Pengujian dan analisis dilakukan untuk mengetahui apakah sistem telah bekerja sesuai perancangan yang telah dilakukan. Pengujian dilakukan per blok sistem kemudian secara keseluruhan. Adapun pengujian yang dilakukan sebagai berikut:

- 1) Pengujian komunikasi serial UART ke PC
- 2) Pengujian hasil *sampling*
- 3) Pengujian pemberian perintah orang yang berbeda
- 4) Pengujian keberhasilan menerima perintah dengan jarak jauh menggunakan *wireless microphone*
- 5) Pengujian keseluruhan sistem

5.1 Pengujian Komunikasi Serial UART ke PC

Pengujian ini dilakukan untuk mengetahui apakah rangkaian mikrokontroler dapat mengirimkan data melalui komunikasi serial UART. Untuk pengujian ini, mikrokontroler mengirimkan data pada terminal komputer menggunakan konfigurasi *baudrate* 9600 bps, 8 bit data, tanpa paritas dan 1 *stop* bit. Data yang dikirimkan merupakan paket data yang berisi pendeteksian modul *EasyVR*.

Arduino uno akan memberikan instruksi sinyal pada pin TX dan RX yang terhubung oleh kabel USB yang kemudian tersambung juga oleh komputer melalui USB. Hasil pengujian ditunjukkan pada Gambar 5.1



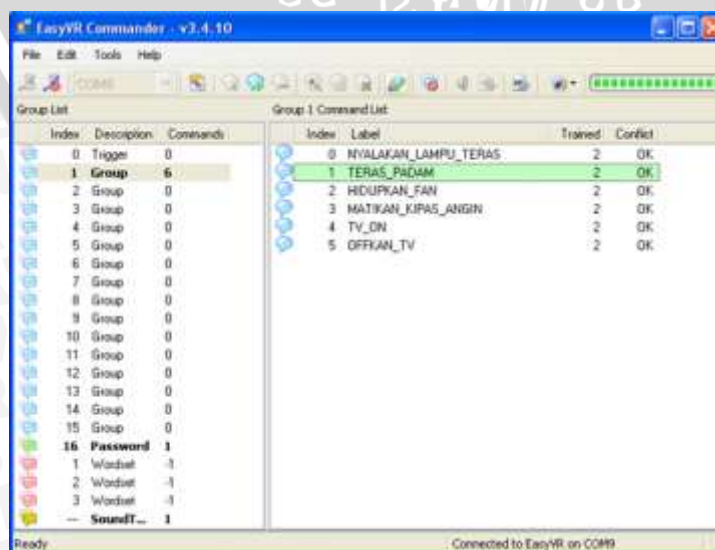
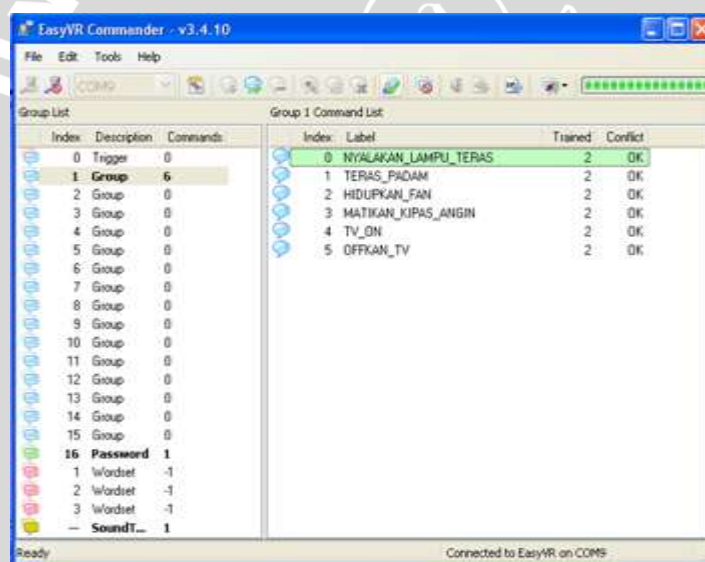
Gambar 5.1 Tampilan Serial Monitor pada *Arduino* untuk pengujian komunikasi serial dengan USB.

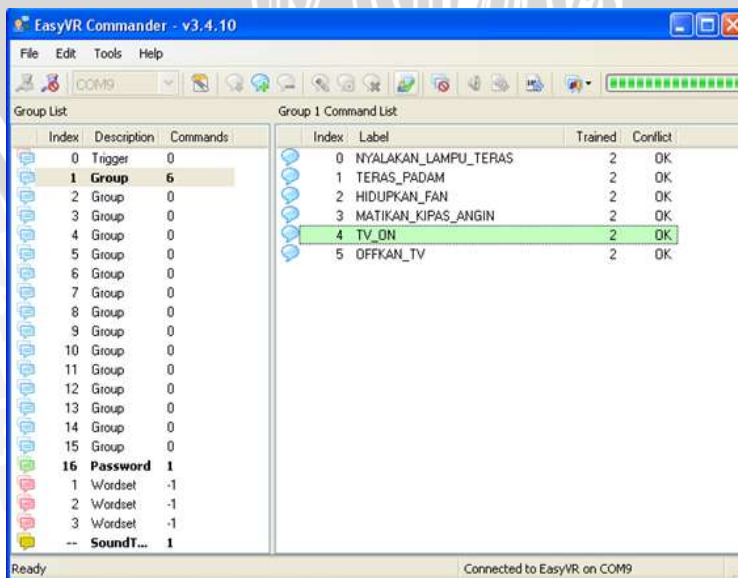
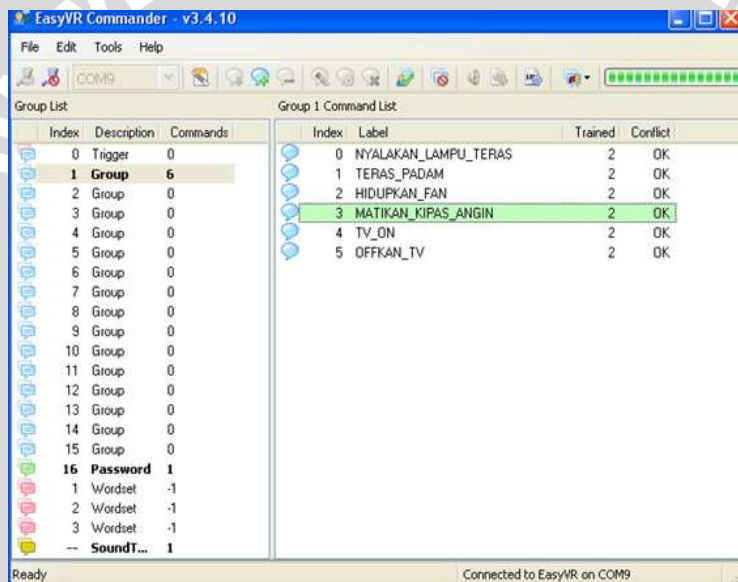
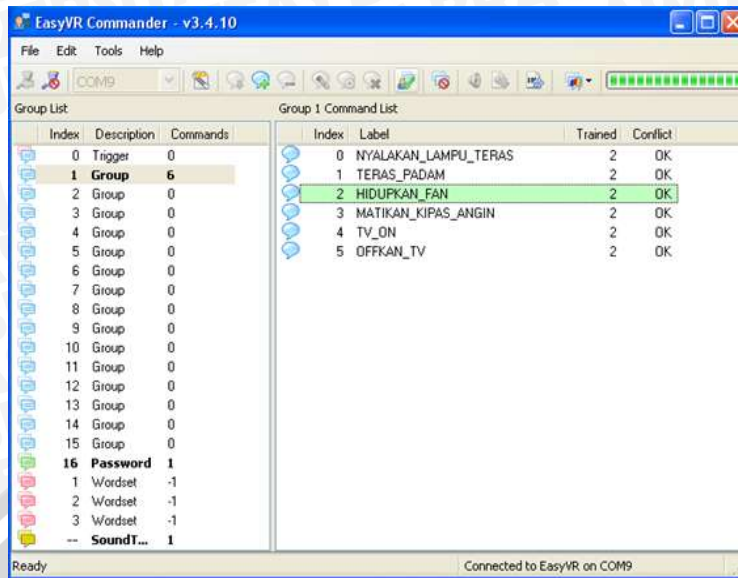
Berdasarkan hasil pengujian yang telah dilakukan, apabila *EasyVR* tidak terdeteksi, maka akan muncul kalimat bertuliskan “*EasyVR not detected*” sebaliknya, apabila terdeteksi maka akan muncul kalimat bertuliskan “*EasyVR detected*”

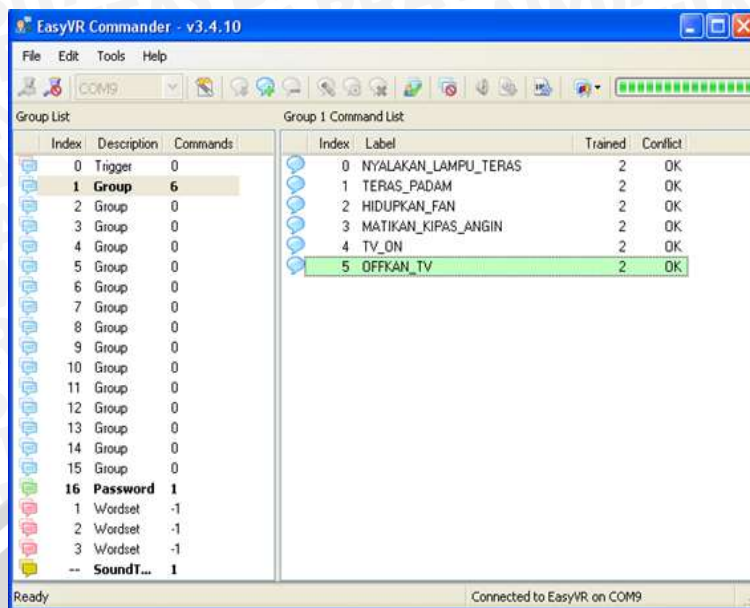
5.2 Pengujian Voice Recognition

5.2.1 Pengujian Hasil Sampling

Tujuan dari pengujian ini adalah untuk menguji apakah hasil *sampling* berupa suara manusia khususnya pengucapan yang telah dituliskan dalam program *EasyVR commander* dapat digunakan dalam penerapan sistem. Pengujian ini dilakukan dengan cara menghubungkan *EasyVR* ke PC atau laptop menggunakan konektor berupa USB to TTL.







Gambar 5.2 Hasil Pengujian *Sampling*.

Gambar 5.2 menunjukkan hasil pengujian *sampling*. Tanda berwarna hijau menunjukkan bahwa kata yang diucapkan sesuai dengan hasil *sampling* yang tersimpan dalam modul *EasyVR*.

5.2.2 Pengujian Tingkat Keberhasilan Pemberian Perintah dengan Variasi Pemberian Perintah dari Orang yang Sama

Tujuan dari pengujian ini adalah untuk membandingkan tingkat keberhasilan *EasyVR* dalam menerima perintah suara apabila diberikan variasi pemberian perintah. Pemberian perintah dilakukan oleh orang yang sama. Terdapat 3 variasi dalam pemberian perintah ini pertama variasi pengucapan sesuai dengan perancangan, kedua variasi pengucapan hanya dengan 2 kata dalam setiap kalimat dan yang ketiga variasi pengucapan dengan menggunakan kode *Alphabet*.

Tabel 5.1 Hasil pengujian pemberian perintah sesuai dengan perancangan

Variasi Perintah	Tingkat Keberhasilan Pengucapan dilakukan sebanyak 10 kali
Nyalakan Lampu Teras	5
Teras Padam	7
Hidupkan Fan	6
Matikan Kipas Angin	5



TV ON	8
OFFkan TV	8
% Keberhasilan	65%

Tabel 5.2 Hasil pengujian pemberian perintah dengan menggunakan 2 kata dalam setiap kalimat

Variasi Perintah	Tingkat Keberhasilan Pengucapan dilakukan sebanyak 10 kali
Lampu Nyala	8
Padam Lampu	7
Kipas ON	9
Mati Kipas	9
TV Hidup	9
OFF TV	9
% Keberhasilan	85%

Tabel 5.3 Hasil pengujian pemberian perintah menggunakan kode *Alphabet*

Variasi Perintah	Tingkat Keberhasilan Pengucapan dilakukan sebanyak 10 kali
A	10
B	10
C	9
D	10
E	10
F	10
% Keberhasilan	98.30%

Dilihat dari hasil pengujian yang terdapat pada tabel 5.1, tabel 5.2 dan tabel 5.3 dapat dibandingkan bahwa, variasi pemberian perintah dengan menggunakan kode *Alphabet* memiliki tingkat keberhasilan yang sangat tinggi, hal ini dipengaruhi oleh *EasyVR* yang dalam setiap pemberian perintah harus sama dengan pengucapan intonasi pada saat suara *disampling*, perintah yang memiliki kalimat yang lebih dari satu kata, biasanya terdapat perbedaan *tone* pengucapan sehingga mengakibatkan kegagalan *EasyVR* dalam mengeksekusi perintah tersebut. Namun, pemberian perintah menggunakan kode *Alphabet* sangatlah tidak *user friendly*

artinya pengguna harus menghafalkan setiap kode dan arti dari kode tersebut. Berikut merupakan arti dari kode *Alphabet* dari A hingga F pada kinerja alat yang telah *disampling* dan dilakukan pengujian.

- A : Lampu Teras Menyala
- B : Lampu Teras Padam
- C : Kipas Angin Menyala
- D : Kipas Angin Padam
- E : Televisi Menyala
- F : Televisi Padam

Berikut Merupakan tabel hasil pengujian besar tegangan yang terdapat pada lampu, kipas angin, dan televisi pada kondisi menyala dan mati ketika pengucapan perintah melalui suara dilakukan

Tabel 5.4 Besar Tegangan Lampu pada Perintah “Nyalakan Lampu Teras”

No.	Perintah Nyalakan Lampu Teras	Keberhasilan Pengucapan 5 Kali	Besar Tegangan Lampu Menyala
1			214.3 Volt AC
2			213.5 Volt AC
3			214.3 Volt AC
4			214.4 Volt AC
5			214.3 Volt AC

Tabel 5.5 Besar Tegangan Lampu pada Perintah “Teras Padam”

No.	Perintah Teras Padam	Keberhasilan Pengucapan 7 Kali	Besar Tegangan Lampu Padam
1			0.004 Volt AC
2			0.004 Volt AC
3			0.002 Volt AC
4			0.003 Volt AC
5			0.004 Volt AC
6			0.003 Volt AC
7			0.003 Volt AC

Tabel 5.6 Besar Tegangan Kipas Angin pada Perintah “Hidupkan Fan”

No.	Perintah Hidupkan Fan	Keberhasilan Pengucapan 6 Kali	Besar Tegangan Kipas Angin Menyala
1			214.5 Volt AC
2			213.5 Volt AC
3			213.5 Volt AC
4			214.5 Volt AC
5			214.6 Volt AC
6			214.5 Volt AC

Tabel 5.7 Besar Tegangan Kipas Angin pada Perintah “Matikan Kipas Angin”

No.	Perintah Matikan Kipas Angin	Keberhasilan Pengucapan 5 Kali	Besar Tegangan Kipas Angin Mati
1			0.003 Volt AC
2			0.004 Volt AC
3			0.001 Volt AC
4			0.003 Volt AC
5			0.002 Volt AC

Tabel 5.8 Besar Tegangan Televisi pada Perintah “TV ON”

No.	Perintah TV ON	Keberhasilan Pengucapan 8 Kali	Besar Tegangan Televisi Menyala
1			214.3 Volt AC
2			214.6 Volt AC
3			213.5 Volt AC
4			214.3 Volt AC
5			214.3 Volt AC
6			214.3 Volt AC
7			214.3 Volt AC
8			214.3 Volt AC

Tabel 5.9 Besar Tegangan Televisi pada Perintah “OFFkan TV”

No.	Perintah OFFkan TV	Keberhasilan Pengucapan 8 Kali	Besar Tegangan Televisi Mati
1			0.002 Volt AC
2			0.004 Volt AC
3			0.003 Volt AC
4			0.004 Volt AC

5	0.003 Volt AC
6	0.003 Volt AC
7	0.003 Volt AC
8	0.002 Volt AC

Tabel 5.10 Besar Tegangan Lampu pada Perintah “Lampu Nyala”

No.	Perintah Lampu Nyala	Keberhasilan Pengucapan 8 Kali	Besar Tegangan Lampu Menyala
1			214.3 Volt AC
2			213.5 Volt AC
3			214.3 Volt AC
4			214.4 Volt AC
5			214.3 Volt AC
6			213.5 Volt AC
7			214.3 Volt AC
8			214.4 Volt AC

Tabel 5.11 Besar Tegangan Lampu pada Perintah “Padam Lampu”

No.	Perintah Padam Lampu	Keberhasilan Pengucapan 7 Kali	Besar Tegangan Lampu Padam
1			0.004 Volt AC
2			0.004 Volt AC
3			0.002 Volt AC
4			0.003 Volt AC
5			0.004 Volt AC
6			0.003 Volt AC
7			0.003 Volt AC

Tabel 5.12 Besar Tegangan Kipas angin pada Perintah “Kipas ON”

No.	Perintah Kipas ON	Keberhasilan Pengucapan 9 Kali	Besar Tegangan Kipas Menyala
1			214.5 Volt AC
2			213.5 Volt AC
3			213.5 Volt AC
4			214.5 Volt AC
5			214.6 Volt AC
6			214.5 Volt AC
7			213.5 Volt AC

8	213.5 Volt AC
9	214.5 Volt AC

Tabel 5.13 Besar Tegangan Kipas angin pada Perintah “Mati Kipas”

No.	Perintah Mati Kipas	Keberhasilan Pengucapan 9 Kali	Besar Tegangan Kipas Mati
1			0.003 Volt AC
2			0.004 Volt AC
3			0.001 Volt AC
4			0.003 Volt AC
5			0.002 Volt AC
6			0.001 Volt AC
7			0.003 Volt AC
8			0.002 Volt AC
9			0.003 Volt AC

Tabel 5.14 Besar Tegangan Televisi pada Perintah “TV Hidup”

No.	Perintah TV Hidup	Keberhasilan Pengucapan 9 Kali	Besar Tegangan Televisi Menyala
1			214.3 Volt AC
2			213.5 Volt AC
3			214.3 Volt AC
4			214.4 Volt AC
5			214.3 Volt AC
6			213.5 Volt AC
7			214.3 Volt AC
8			214.4 Volt AC
9			214.3 Volt AC

Tabel 5.15 Besar Tegangan Televisi pada Perintah “OFF TV”

No.	Perintah OFF TV	Keberhasilan Pengucapan 9 Kali	Besar Tegangan Televisi Mati
1			0.004 Volt AC
2			0.004 Volt AC
3			0.002 Volt AC
4			0.003 Volt AC
5			0.004 Volt AC
6			0.003 Volt AC
7			0.003 Volt AC
8			0.003 Volt AC

Tabel 5.16 Besar Tegangan Lampu pada Perintah "A"

No.	Perintah A	Keberhasilan Pengucapan 10 Kali	Besar Tegangan Lampu Menyala
1			214.3 Volt AC
2			213.5 Volt AC
3			214.3 Volt AC
4			214.4 Volt AC
5			214.3 Volt AC
6			213.5 Volt AC
7			214.3 Volt AC
8			214.4 Volt AC
9			214.3 Volt AC
10			213.5 Volt AC

Tabel 5.17 Besar Tegangan Lampu pada Perintah "B"

No.	Perintah B	Keberhasilan Pengucapan 10 Kali	Besar Tegangan Lampu Padam
1			0.004 Volt AC
2			0.004 Volt AC
3			0.002 Volt AC
4			0.003 Volt AC
5			0.004 Volt AC
6			0.003 Volt AC
7			0.003 Volt AC
8			0.002 Volt AC
9			0.003 Volt AC
10			0.004 Volt AC

Tabel 5.18 Besar Tegangan Kipas Angin pada Perintah "C"

No.	Perintah C	Keberhasilan Pengucapan 9 Kali	Besar Tegangan Kipas Menyala
1			214.5 Volt AC
2			213.5 Volt AC
3			213.5 Volt AC
4			214.5 Volt AC
5			214.6 Volt AC

6	214.5 Volt AC
7	213.5 Volt AC
8	213.5 Volt AC
9	214.5 Volt AC

Tabel 5.19 Besar Tegangan Kipas Angin pada Perintah “D”

No.	Perintah D	Keberhasilan Pengucapan 10 Kali	Besar Tegangan Kipas Mati
1			0.003 Volt AC
2			0.004 Volt AC
3			0.001 Volt AC
4			0.003 Volt AC
5			0.002 Volt AC
6			0.001 Volt AC
7			0.003 Volt AC
8			0.002 Volt AC
9			0.003 Volt AC
10			0.002 Volt AC

Tabel 5.20 Besar Tegangan Televisi pada Perintah “E”

No.	Perintah E	Keberhasilan Pengucapan 10 Kali	Besar Tegangan Televisi Menyala
1			214.3 Volt AC
2			213.5 Volt AC
3			214.3 Volt AC
4			214.4 Volt AC
5			214.3 Volt AC
6			213.5 Volt AC
7			214.3 Volt AC
8			214.4 Volt AC
9			214.3 Volt AC
10			214.4 Volt AC

Tabel 5.21 Besar Tegangan Televisi pada Perintah “F”

No.	Perintah F	Keberhasilan Pengucapan 10 Kali	Besar Tegangan Televisi Menyala
1			0.004 Volt AC
2			0.004 Volt AC
3			0.002 Volt AC

4	0.003 Volt AC
5	0.004 Volt AC
6	0.003 Volt AC
7	0.003 Volt AC
8	0.003 Volt AC
9	0.003 Volt AC
10	0.003 Volt AC

Dilihat dari data hasil pengujian besar rata-rata tegangan agar Lampu, Kipas Angin dan Televisi Menyala berada pada kisaran 214 Volt pada sumber AC, sedangkan besar rata-rata tegangan pada saat Lampu, Kipas Angin, dan Televisi mati sebesar 0.004 Volt AC.

5.2.3 Pengujian Pemberian Perintah dari Orang yang Berbeda

Tujuan dari pengujian ini adalah untuk mengetahui tingkat keberhasilan *EasyVR* dalam menerima perintah suara dari orang yang berbeda. Perintah diberikan oleh 5 orang berjenis kelamin laki-laki dan perempuan dengan berbagai etnis dengan rentang usia 19 hingga 25 tahun. Setiap pemberian perintah dilakukan 10 kali per perintah dengan jarak maksimal 30 meter dari antenna pemancar, dan 10 cm dari sensor *michrophone* yang dipegang pada setiap orang. Pengujian dilakukan dengan kondisi ideal atau *noise* yang sangat kecil dengan ukuran luas rumah 45m² satu lantai.

Tabel 5.22 Hasil pengujian pemberian perintah dari orang yang berbeda

Orang ke-	Jumlah Keberhasilan Pemberian Perintah Suara						% Keberhasilan
	Nyalakan Lampu Teras	Teras Padam	Hidupkan Fan	Matikan Kipas Angin	TV ON	OFFkan TV	
1	5	7	8	5	8	7	66,7%
2	4	8	8	5	7	8	66,7%
3	4	5	6	4	6	5	50%
4	4	6	5	5	7	5	53,3%
5	3	5	5	4	6	6	48,3%
% Rata-rata Keberhasilan							50%

Tabel 5.22 menunjukkan berapa kali perintah harus diucapkan agar *EasyVR* mampu mengolah perintah yang diberikan agar sama dengan hasil samping. Hasil pengujian menunjukkan tingkat keberhasilan menerima perintah suara dari kondisi yang sudah ditentukan dan pengucapan perintah suara yang memiliki

variasi sama dengan sample yang direkam mencapai 50% kegagalan dalam pengolahan perintah suara disebabkan oleh pengucapan tutur kata yang intonasinya berbeda dengan hasil sampling suara, terutama pada kalimat yang memiliki 3 kata, pengucapan dengan intonasi yang berbeda dengan hasil *sampling* suara menyebabkan *EasyVR* tidak dapat mendeteksi suara sehingga suara menjadi tidak dapat dikenali.

5.3 Pengujian Keberhasilan Menerima Perintah Dengan Jarak Jauh Menggunakan *Wireless Michrophone*

Tujuan dari pengujian ini adalah untuk mengetahui tingkat keberhasilan pengucapan perintah jarak jauh. Pengujian dilakukan dalam sebuah rumah oleh orang yang sama dengan menggunakan variasi perintah yang sesuai dengan perancangan selama pengujian ini berlangsung, dengan luas rumah sebesar 45m² satu lantai, memiliki sekat 2 kamar tidur, dengan melakukan pengucapan sebanyak 10kali, jarak peletakan *michrophone* ditentukan sejauh 10cm. Pengujian dilakukan hanya dari dalam rumah tidak dari luar rumah.

Tabel 5.23 Hasil Pengujian keberhasilan menerima perintah dengan jarak jauh menggunakan *wireless michrophone*

	Jarak Pemberian Perintah							
	5 m	10 m	15 m	20 m	25 m	30m	35m	40m
Nyalakan Lampu Teras	7	6	6	5	5	4	0	0
Teras Padam	8	8	7	6	6	5	0	0
Hidupkan Fan	7	8	8	7	6	5	0	0
Matikan Kipas Angin	5	5	6	5	6	4	0	0
TV ON	9	8	8	7	8	7	0	0
OFFkan TV	7	7	6	7	7	6	0	0

Tabel 5.23 menunjukkan tingkat keberhasilan menerima perintah dengan jarak jauh menggunakan *wireless michrophone*, hasil pengujian menunjukkan tingkat

keberhasilan pengucapan perintah dari jarak jauh, terlihat bahwa jarak pemberian perintah jarak jauh maksimal sejauh 30 m.

5.4 Pengujian Keseluruhan Sistem

Pengujian Keseluruhan Sistem secara Keseluruhan dibagi menjadi Dua bagian yaitu:

1. Pengujian melalui serial monitor dalam program arduino
2. Pengujian respon peralatan rumah tangga setelah menerima perintah dalam satuan waktu

5.4.1 Pengujian Melalui Serial Monitor pada Program *Arduino*

Tujuan dari pengujian ini adalah untuk mengetahui apakah board *Arduino UNO* mampu menerima data ASCII dari *EasyVR* dan mengeksekusi data tersebut. Pengujian ini dilakukan dengan cara menghubungkan *Arduino UNO* yang sudah terpasang dengan perancangan sistem secara keseluruhan dengan PC atau laptop melalui kabel USB. Hasil pengujian dapat dilihat melalui serial monitor yang terdapat pada program *Arduino*.



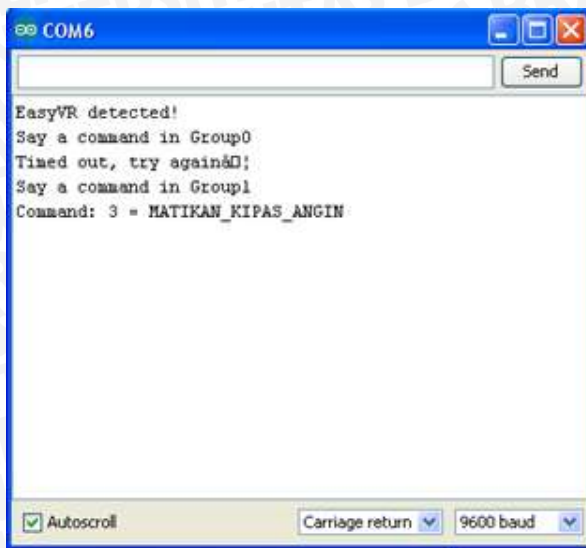
```
COM6
EasyVR detected!
Say a command in Group0
Timed out, try again!
Say a command in Group1
Error 11
Say a command in Group1
Command: 0 = NYALAKAN_LAMPU_TERAS
Say a command in Group1
Error 11
Say a command in Group1
Command: 1 = TERAS_PADAM
Say a command in Group1
```

(a)



```
COM6
EasyVR detected!
Say a command in Group0
Timed out, try again!
Say a command in Group1
Command: 2 = HIDUPKAN_FAN
Say a command in Group1
```

(b)



(c)



(d)



(e)

Gambar 5.3 Menunjukkan tampilan serial Monitor untuk pengujian peralatan listrik (a) Lampu Teras menyala dan padam, (b) Kipas Menyala, (c) Kipas padam, (d) TV menyala, (e) Tv padam

Gambar 5.3 menunjukkan respons dari setiap kata yang diucapkan. Dari hasil pengujian terlihat bahwa respon alat mampu menerima perintah berupa suara. Terbukti dari setiap pengucapan perintah sesuai dengan *command* yang diberikan.

5.4.2 Pengujian Respon Peralatan Rumah Tangga Setelah Menerima Perintah Dalam Satuan Waktu

Tujuan dari pengujian ini adalah untuk mengetahui berapa lama respon alat melakukan perintah suara yang diberikan. Pengujian dilakukan dengan cara manual yaitu dengan menggunakan jam digital sebagai pewaktu. Perhitungan waktu dimulai setelah perintah selesai diucapkan.

Tabel 5.24 Hasil Pengujian rata-rata waktu respon peralatan listrik setelah menerima perintah

Perintah	Waktu Respon
Nyalakan Lampu Teras	≤ 2 detik
Teras Padam	≤ 2 detik
Hidupkan Fan	≤ 2 detik
Matikan Kipas Angin	≤ 2 detik
Kipas Padam	≤ 2 detik
TV ON	≤ 2 detik
OFFkan TV	≤ 2 detik

Tabel 5.24 menunjukkan bahwa rata-rata waktu respon dari setelah perintah diucapkan sampai peralatan listrik melakukan perintah adalah kurang dari sama dengan 2 detik. Hal ini disebabkan berjalannya proses ketika *microphone* menerima perintah yang kemudian dicocokkan oleh *EasyVR* apakah perintah yang diberikan sesuai dengan *sample*. Jika sesuai, *EasyVR* mengirim perintah ke Arduino UNO yang kemudian akan mengakses *relay*.

BAB VI

PENUTUP

6.1 Kesimpulan

Dari hasil perancangan dan pengujian yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

- 1). Metode pengambilan *sample* suara pada modul *EasyVR* dilakukan sebanyak 2 kali menggunakan *EasyVR Commander* dengan variasi pengucapan relatif sama pada setiap kata.
- 2). Modul *EasyVR* memiliki tingkat keberhasilan sekitar 50% jika menerima perintah suara dari orang yang berbeda dengan variasi pemberian perintah 2 hingga 3 suku kata.
- 3). Efektifitas penggunaan modul *EasyVR* dalam hal pengenalan suara pemberian perintah efektif jika perintah yang digunakan hanya berupa satu suku kata, tingkat keberhasilan sebesar 98.30%.
- 4). Kesalahan yang sering terjadi pada proses pengucapan perintah adalah perbedaan intonasi suara yang pada saat pemberian perintah tidak sesuai dengan pada saat proses *sampling* suara, menyebabkan perintah tidak dikenali.

6.2 Saran

Sistem pengenalan suara sangat bergantung kepada variasi *sample* yang direkam. Oleh karena itu, pemberian perintah lebih baik hanya menggunakan satu kata atau berupa kode, sehingga variasi pengucapan perintah yang sesuai dengan *sample* yang direkam dapat lebih mudah diingat dibandingkan apabila menggunakan pengucapan perintah yang lebih dari satu kata.

DAFTAR PUSTAKA

- Atmel, 2007. ATMEGA16/ATMEGA16L, 8-bit AVR with 8 kbytes in System Programmable Flash. www.atmel.com/literatur. Diakses tanggal 8 Maret 2014.
- Ardi, Jatra Kurnia. 2014. *Implementasi Sistem Voice Recognition Pada Robot Pemindah Objek Sebagai Sistem Navigasi*. Malang: Skripsi Jurusan Teknik Elektro FT-UB.
- Arduino, 2009. *Datasheet Arduino UNO*.
<http://arduino.cc/en/Main/arduinoBoardUNO>. Diakses tanggal 24 Februari 2014.
- Kadir, Abdul., Heriyanto. 2005. *Algoritma Pemrograman C++*. Yogyakarta: Andi Offset.
- Malvino, Albert Paul. 1985. *Prinsip-Prinsip Elektronika Jilid 1*. Jakarta: Erlangga
- Putra, Darma. 2009. *Sistem Biometrika*. Yogyakarta: Andi.
- Rabiner, LR Juang, BH. 1993. *Fundamental of Speech Recognition*. New Jersey: Prentice Hall.
- Tigal, 2012. Manual User *EasyVR 2.0*. <http://www.veear.eu/>. Diunduh tanggal 3 Maret 2014. Austria: TIGAL KG.
- Warren, Hioki. 2000. *Telecommunications*. New Jersey: Prentice Hall.
- Woolard, Barry. 2003. *Practical Electronics*. United Kingdom: McGraw-Hill Book Company.

LAMPIRAN 1

FOTO ALAT

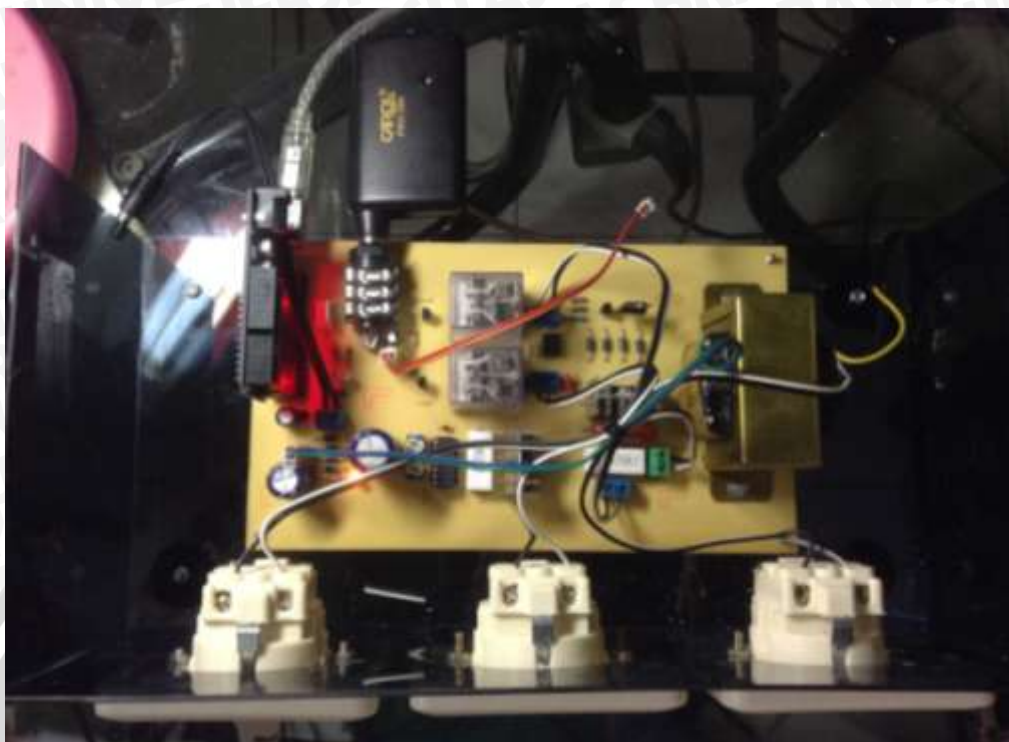




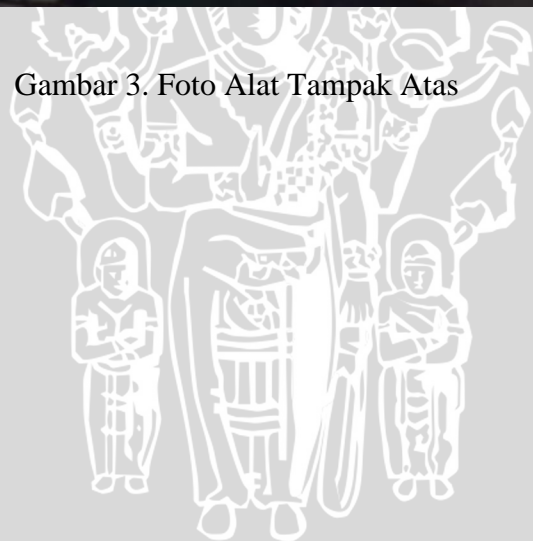
Gambar 1. Foto Alat Tampak Depan



Gambar 2. Foto Alat Tampak Belakang



Gambar 3. Foto Alat Tampak Atas



LAMPIRAN 2

LISTING PROGRAM



```
int LED1=2;

int LED2=3;

int LOAD1=4; //lampu

int LOAD2=5; //kipas angin

int LOAD3=6; //televisi

#if defined(ARDUINO)&&ARDUINO >= 100 //define arduino uno reserve words

#include "Arduino.h"

#include "SoftwareSerial.h" // software serial

SoftwareSerial port(12,13); // TX dan RX (komunikasi serial) virtuil software serial.

#else // Arduino 0022 – menggunakan modified NewSoftSerial

#include "WProgram.h"

#include "NewSoftSerial.h"

NewSoftSerial port(12,13);

#endif

#include "EasyVR.h"

EasyVR easyvr(port); //reserve word

//Groups dan Commands

enum Groups

{

//GROUP_0 = 0,

GROUP_1 = 1,

};

enum Group0

{

G0_ARDUINO = 0,

};

enum Group1
```

```
{  
  G1_NYALAKAN_LAMPU_TERAS = 0, //G1 maksudnya group 1  
  G1_TERAS_PADAM = 1,  
  G1_HIDUPKAN_FAN = 2,  
  G1_MATIKAN_KIPAS_ANGIN = 3,  
  G1_TV_ON = 4,  
  G1_OFFKAN_TV = 5,  
};
```

```
EasyVRBridge bridge;
```

```
int8_t group, idx;
```

```
void setup()
```

```
{  
  pinMode(LED1, OUTPUT); //pinmode sebagai deklarasi  
  pinMode(LED2, OUTPUT);  
  pinMode(LOAD1, OUTPUT);  
  pinMode(LOAD2, OUTPUT);  
  pinMode(LOAD3, OUTPUT);
```

```
  digitalWrite(LOAD1, LOW); //kondisi awal mati
```

```
  digitalWrite(LOAD2, LOW);
```

```
  digitalWrite(LOAD3, LOW);
```

```
  digitalWrite(LED1, HIGH);
```

```
  digitalWrite(LED2, HIGH);
```

```
  // bridge mode?
```

```
  if (bridge.check())
```

```
{
cli();
bridge.loop(0, 1, 12, 13);
}
// run normally
Serial.begin(9600); //inisialisasi speed komunikasi dengan laptop dan arduino
port.begin(9600); //

if (!easyvr.detect()) //not detected
{
Serial.println("EasyVR not detected!"); //
for (;;);
}
easyvr.setPinOutput(EasyVR::IO1, LOW);
Serial.println("EasyVR detected!");
easyvr.setTimeout(5); //5 detik (out of time)
easyvr.setLanguage(EasyVR::ENGLISH);

group = EasyVR::TRIGGER; //start group (customize)
}

void action();

void loop()
{
easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)
Serial.print("Say a command in Group");
Serial.println(group);
easyvr.recognizeCommand(group);
```

```
do
{
// can do some processing while waiting for a spoken command
}
while (!easyvr.hasFinished()); //
```

```
easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off
```

```
if (idx >= 0)
```

```
{
```

```
// built-in trigger (ROBOT)
```

```
// group = GROUP_X; &lt;- jump to another group X
```

```
return;
```

```
}
```

```
if (idx >= 0) //syarat variabel
```

```
{
```

```
// print debug message
```

```
uint8_t train = 0;
```

```
char name[32];
```

```
Serial.print("Command: ");
```

```
Serial.print(idx);
```

```
if (easyvr.dumpCommand(group, idx, name, train))
```

```
{
```

```
Serial.print(" = ");
```

```
Serial.println(name);
```

```
}
```

```
else
```

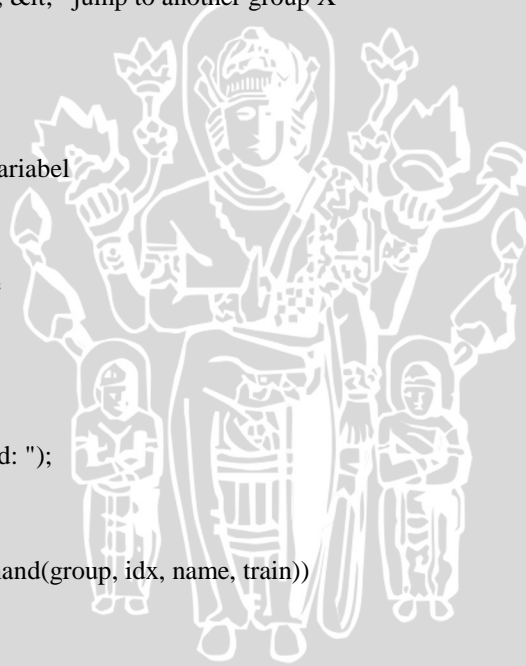
```
Serial.println();
```

```
easyvr.playSound(0, EasyVR::VOL_FULL);
```

```
// perform some action
```

```
action();
```

UNIVERSITAS BRAWIJAYA




```
}  
else // errors or timeout  
{  
    if (easyvr.isTimeout())  
        Serial.println("Timed out, try again...");  
    int16_t err = easyvr.getError();  
    if (err >= 0)  
    {  
        Serial.print("Error ");  
        Serial.println(err, HEX);  
    }  
  
    group = GROUP_1;  
}  
  
void action()  
{  
    switch (group) //  
    {  
        // case GROUP_0:  
        // switch (idx)  
        // {  
        // case G0_ARDUINO:  
        // write your action code here  
        // group = GROUP_1; //&lt;- or jump to another group X for composite commands  
        // break;  
        // }  
        // break;  
        case GROUP_1:  
            switch (idx)
```



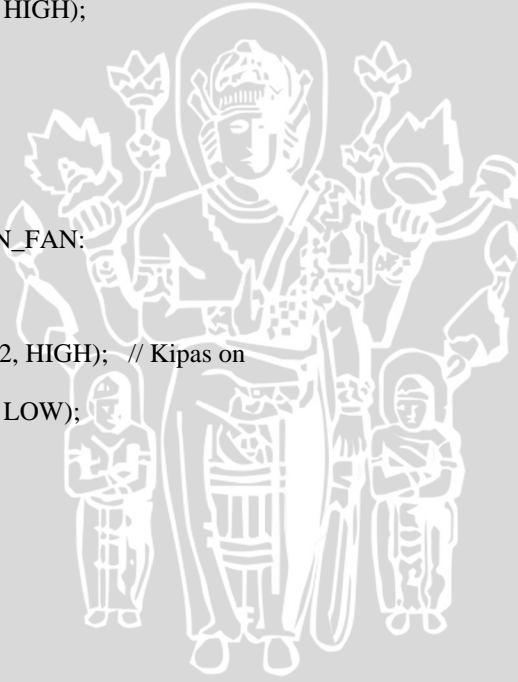
```
{
  case G1_NYALAKAN_LAMPU_TERAS:
    Serial.print(" ");
    digitalWrite(LOAD1, HIGH); // Lampu on
    digitalWrite(LED1, LOW);
    delay(1000);
    break;

  case G1_TERAS_PADAM:
    Serial.print(" ");
    digitalWrite(LOAD1, LOW); // Lampu off
    digitalWrite(LED1, HIGH);
    delay(1000);
    break;

  case G1_HIDUPKAN_FAN:
    Serial.print(" ");
    digitalWrite(LOAD2, HIGH); // Kipas on
    digitalWrite(LED2, LOW);
    delay(1000);
    break;

  case G1_MATIKAN_KIPAS_ANGIN:
    Serial.print(" ");
    digitalWrite(LOAD2, LOW); // Kipas off
    digitalWrite(LED2, HIGH);
    delay(1000);
    break;

  case G1_TV_ON:
```



```
Serial.print(" ");  
digitalWrite(LOAD3, HIGH); // tv on  
delay(1000);  
break;  
  
case G1_OFFKAN_TV:  
  
Serial.print(" ");  
digitalWrite(LOAD3, LOW); // tv off  
delay(1000);  
break;  
  
}  
break;  
  
}  
}
```



LAMPIRAN 3

DATASHEET

